DETECTING ANOMALOUS TRAFFIC USING STATISTICAL PROCESSING AND SELF-ORGANIZING MAPS

Paola Baldassarri, Anna Montesanto and Paolo Puliti
Department of Electronics, Artificial Intelligence and Telecommunications
Polytechnic University of Marche, Via Brecce Bianche 1, 60131 Ancona, Italy

Keywords: Intrusion Detection System, Anomaly Detection, Statistical processing, Self-Organizing Maps.

Abstract: The main idea of the present work is to create a system able to detect intrusions in computer networks. For this purpose we propose a novel intrusion detection system (IDS) based on an anomaly approach. We analyzed the network traffic from (outbound traffic) and towards (inbound traffic) a victim host through another host. Besides we realized an architecture consisted of two subsystems: a statistical subsystem and a neural networks based subsystem. The first elaborates chosen features extracted from the network traffic and it allows determining if an attack occurs through a preliminary visual inspection. The neural subsystem receives in input the output of the statistical subsystem and it has to indicate the status of the monitored host. It classifies the network traffic distinguishing the background traffic from the anomalous one. Moreover the system has to be able to classify different instances of the same attack in the same class, distinguishing in a completely autonomous way different typology of attack.

1 INTRODUCTION

One of the main research fields in the information security concerns the realization of systems able to identify intrusions. In order to meet this challenge, the Intrusion Detection Systems (IDS) are designed to protect the availability, confidentiality and integrity of critical networked information systems (Labib and Vemuri, 2004). Existing intrusion detection techniques can be subdivided in two main categories: misuse detection (Lee et al., 1999; Vigna and Kemmerer, 1998) and anomaly detection (Gosh et al., 1998; Valdes and Anderson, 1995). Misuse detection techniques compare streams extracted from the network traffic with signatures of known attacks and so they indicate if an intrusion occurs when there is a match. In this case the system is able to detect only known attacks. Anomaly detection techniques create a profile of normal behaviour of a subject and/or a system (normal profile), then compare the observed behaviour with the normal profile, and signal an intrusion when two behaviours significantly deviate. Therefore, for the anomaly detection techniques is necessary to create a normal profile in order to consider a deviation from the normal behaviour as symptom of an intrusion. The first step is to establish which a normal behaviour is, and subsequently to indicate the statistical features that describe it. This phase concerns a wide range of aspects: from the system call of an operating system to a list of open files, from the time of use of the CPU to all the parameters regarding a TCP/IP connection and so on. In literature the statistical modelling followed with classical or neural network classification has been utilized in anomaly intrusion detection systems (Cabrera et al., 2000; Ye et al., 2002; Zhang et al., 2001).

In this ambit we propose a new IDS based on an anomaly detection approach. The system will observe the network traffic of a host, probably a web server, a mail server or a remote authentication server. Our IDS does not reside on the monitored server, but in another computer for this purpose dedicated. So, the IDS does not depend by the operating system of the monitored server, guarantying integrity and robustness, also in the case of attacked and compromised system. The detection system has to verify if the monitored host is generating statistically different network traffic. For this purpose our idea is to combine two ap-

Baldassarri P., Montesanto A. and Puliti P. (2007).
DETECTING ANOMALOUS TRAFFIC USING STATISTICAL PROCESSING AND SELF-ORGANIZING MAPS.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 74-79
DOI: 10.5220/0002123500740079

proaches: a statistical approach and neural networks based approach. The system consists of two subsystems. The statistical subsystem, also named ”discriminator”, considers sliding time windows from which the network traffic is observed. Moreover the output of the statistical subsystem is the input of the neural subsystem also named ”decisional motor”, which has to indicate the status of the system. In particular we used the Self-Organizing Maps (SOM) (Kohonen, 2001) based on an unsupervised learning which independently organize the input patterns into various classes. The SOM have been used in a lot of works concerning the IDS. In (DeLooze, 2006) the author considered several SOM using specific feature sets for each attack type, improving the value of the detection rate and reducing the false alarm rate. Depren et al. (Depren et al., 2005) propose a hybrid IDS that consists of 3 modules. Besides the first module consists of 3 specific SOM, each of them operate on different protocols (TCP, UDP and ICMP). In our IDS the ”decisional motor” is a two-tier architecture. In the first layer each SOM individually operates on a different feature extracted from the network traffic. Then, the last layer combines the results of the first layer.

2 DESCRIPTION OF IDS

Our system can be classified as an anomaly based IDS. It does not require an ”a priori knowledge” of the attacks, neither a constant updating of the attacks signatures, since it would find the ”bad behaviours” on the base of a ”normality” description. The system observes the in-bound and out-bound traffic of a server that provides network services. As the figure 1 shows, the system consisted of two subsystems: a statistical subsystem and a neural networks based subsystem, both described in the following paragraphs.

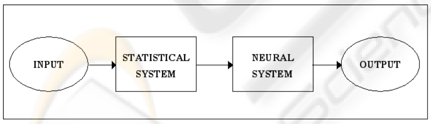

Figure 1: Block diagram of the implemented IDS.

2.1 The Statistical Subsystem

The aim of the statistical subsystem is to capture a statistical correlation among a flow of packets. In the same way, events as ”strange increase of flow towards a particular port” or ”anomalous distribution of the flow of packets as regards the normal”, or again ”packets assigned to different port, coming from the same port of the same remote host”, can be recognized. In order to analyze the network traffic and then to extract its characteristic features we considered time windows of 60 seconds (corresponding to 1 period). This choice is also related to the dataset that we used for the experimental phase: the 1999 DARPA/MIT Lincoln Laboratory intrusion detection evaluation dataset (IDEVAL) (Haines et al., 1999).

We filter the in-bound and the out-bound traffic of the monitored server, basing on its IP address. From the available features, 8 were selected for use in the system: 4 for the input packets, and 4 for the output packets. For the input packets, we considered:

1. Source IP Address (SourIP) for the TCP, UDP and ICMP protocols;
2. Source Port (SourPort) for the TCP and UDP protocols;
3. Combination of Source IP Address and Source Port (IP:Port) for the TCP and UDP protocols;
4. Total number of packets (InNPkt) for the TCP, UDP and ICMP protocols;

While for the output packets we considered:

1. Destination IP Address (DesIP) for the TCP, UDP and ICMP protocols;
2. Source Port (SourPort) for the TCP and UDP protocols;
3. Combination of Destination IP Address and Source Port (IP:Port) for the TCP and UDP protocols;
4. Total number of packets (OutNPkt) for the TCP, UDP and ICMP protocols;

For the 8 features we calculate the occurrences of all different data observed in 1 period. For example for the SourIP, how many times in 1 period the same Source IP address sends a packet to the monitored server. Or again, how many packets the monitored server receives in 1 period, and so on.

Then, in order to elaborate a continuous flow of data, the statistical system uses overlapped sliding windows including 5 periods and each window is processed as follows. For the first period, with reference to each feature, the occurrences of the data have to be ordered according to a decreasing sorting. Since, at any one time we considered 5 periods, once sorted the data of the first period, the data of the second period are added as follows. In the case of a packet sent by a yet observed IP address, its occurrence will be increased of 1. Instead, in the case of an unseen IP address, this value with its occurrence is added to the queue. Each data is encoded using integer value (the first IP address has value 1, the second IP address has value 2, and so on). The data are updated in a window of 5 periods departing from the ordered data of the first period. The new data with their occurrences are inserted at the end, obtaining the new graph. The same spiel can be given for the third period, the fourth period and finally for the last period, belonging to the same sliding window of 5 periods. Then, in order to characterize the trend of data of 5 periods we introduce the ”first momentum”. The ”first momentum” equation is represented as follows:

$$\mathrm{m}_1 = \frac{\sum_{x=1}^{k} x f(x)}{\sum_{x=1}^{k} f(x)} \tag{1}$$

where x are the data on the ”X-axis” encoded as integer value, $f(x)$ corresponds to the occurrence of x, and finally k represents the number of different data. Consequently for each overlapped window of 5 periods we determine the m_1, and so each window is represented by an only m_1 value. Resuming, the first window consists of ”1,2,3,4,5” periods obtained departing from the decreasing sorting of the 1 period. The second window considers the ”2,3,4,5,6” periods obtained departing from the decreasing sorting of the 2 period, and so on. It is necessary to point out that these two windows (”1,2,3,4,5” periods and ”2,3,4,5,6” periods respectively) are contemporary but separately processed.

Concerning the features (InNPkt and OutNPkt) we did not determine the value of m_1, but we calculated the mean value for each sliding window of 5 periods. So, for each sliding window we obtained two mean values: one for the in-bound packets and the other for the out-bound packets. All values are stored in two different files: one for the in-bound packets, and the other for the out-bound packets.

2.2 The Neural Subsystem

The neural subsystem is also named ”Decisional Motor”, because it has to indicate if the monitored server has a normal behaviour or if an attack occurs. It has a two-tier architecture consisted of SOM and based on an unsupervised learning. As the figure 2 shows, the first layer has 4 elements, corresponding to structurally identical SOM. They simultaneously and separately process the files: the first SOM (named IP:Port network) processes the IP:Port data, the second SOM (named Port network) processes the Port data, the third SOM (named IP network) processes the IP data, and finally the last SOM (named NPkt network) processes the number of packets data. The weight vectors of each network are two-dimensional, since the input data have two components: one referring to the input packets and the other referring to the output packets. After the training phase, each node of the networks identifies a different class. So, the networks would classify the normal traffic into the same class, while in the case of anomalous behaviour they would classify the same typology of attack in the same class. Each network of the first layer can independently detect an anomalous situation, and the second layer on the base of the most complete information has to indicate or not if an attack occurs. The second layer consisted of only one SOM receives in input the 4 winner nodes of the first layer and so, it classifies the traffic basing on the comprehensive data related to all the features. So, the second layer has to reduce possible false alarms or has to put in evidence an anomalous situation. The winner node of the second layer indicates the class of belonging of the traffic. The class can characterize the normal traffic or a particular attack.

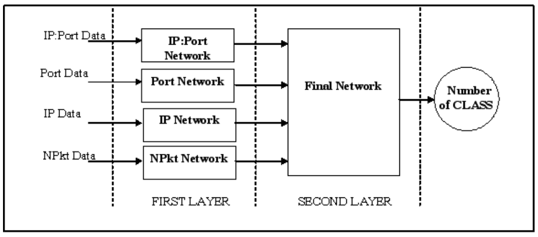

Figure 2: The Neural Networks Architecture.

3 EXPERIMENTAL RESULTS

In the development of an experimental project it is necessary a complete and a wide dataset. Our choice was oriented towards the on line available 1999 Defense Advanced Research Projects Agency (DARPA) Intrusion Detection Evaluation dataset of Lincoln Laboratory. This dataset is actually used in a high number of other works related to the development of IDS (Mahoney and Chan, 2003). The dataset is produced in order to recreate the normal or background traffic and a given number of attacks in a purposely dedicated network. The background traffic was generated considering both the characteristics of the dataflow observed near to the Hanscom Air Force Base military American base, and the statistics on the traffic reaped from other basis. The attacks and the hacking code are extracted from Internet or autonomously developed. They are simultaneously executed with the background traffic. In the dataset were established four categories of attacks: Denial of Service (DoS), probe, remote-to-local (R2L), and user-to-root (U2R). Within these categories they ran several select instances of attacks. Even if these attacks are not comprehensive of the category of attacks, they can be considered as samples from the attack space within the category (Gosh et al., 2000). Lincoln considered five consecutive weeks, producing a dataset containing the network traffic of a rather long period. In particular, the weeks consisted of 5 days (form Monday to Friday) and besides the days consisted of 22 hours (from the 8:00 a.m. to the successive 6:00 a.m.). There are two weeks (the first and the third) with only background traffic, while the attacks are concentrated in the second (background traffic with a low rate of attacks) and in particular in fourth and in fifth (background traffic with a high rate of attacks) weeks.

During the experimental phase, we analyzed the network traffic from and towards a victim host. For this purpose, we chose Pascal.eyrie.af.mil with IP address: 172.16.112.50, with Solaris 2.5 operating system, on an UltraOne system. Pascal allows to the users to read E-mail, to navigate on the Web through Lynx, or to use other services as: Telnet, SMTP, Ssh e FTP.

In the successive subparagraphs we subdivided the results obtained after the statistical processing and the final results related to the neural subsystem.

3.1 Results of the Statistical Subsystem

The figure 3 represents the trend of the m_1 value related to the IP:Port data. We have to remember that the m_1 value is calculated on the overlapped sliding windows of 5 periods (for example 0-4 periods, 1-5 periods and so on). In particular we represent the trend of this feature considering a day of normal traffic (the second day of the first week). The other three features are not represented since they show similar trend. The graph shows a peak (corresponding to 8 value) related to the 194-198 interval. Analyzing the content of the data file, we noted that the period 194 has a different behaviour from the others: this is due to the fact that all the IP addresses are communicating on the same TCP 25 Port. Summarizing, the "IP:Port graph" has rather high values where a consistent number of IP addresses are observed or however different Ports are considered.

An attack will be detected through some observed parameters. For example a kind of attack could be detected by the "IP:Port graph" and the "IP graph", while another kind of attack could be detected by the "IP:Port graph" and "Port graph". This depends on the implicated parameters: some attacks involve a high number of different IP addresses, others a different number of Port from a limited number of different IP addresses, and so on. Similarly an anomalous situation could be noted by the only graphs related to the input packets and/or the output packets. That is why we contemporary considered more than one characteristic. Moreover the neural system has to indicate if there is a normal traffic or which attack occurs through the statistical characteristics they highlighted the attack.

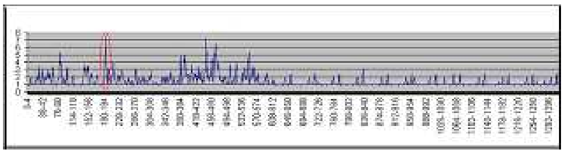

Figure 3: "IP:Port graph" related to the background traffic observed in the input packets (X-axis indicates the interval corresponding of 5 periods, Y-axis indicates the m_1 value).

As example, in figure 4 we showed only the output of the "discriminator" in consequence of a DoS attack (Mailbomb attack). We showed only the IP:Port graph, considering that the other features (Port and NPkt) that detect the attack show a similar trend.

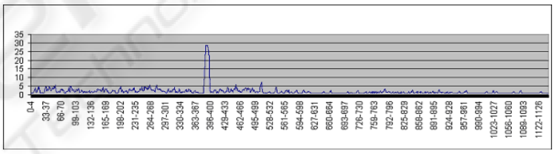

Figure 4: "IP:Port Graph" related to traffic with attacks observed in the input packets (X-axis indicates the interval corresponding of 5 periods, Y-axis indicates the m_1 value).

The "Mailbomb" attack is based on the sending of a high number of e-mails against a mail server in order to crash the system. A typical "Mailbomb" attack occurs through the mailing of 10000 messages from some users (10 Mbyte of data for each user). In IDEVAL there are 3 different instances of this attack against Pascal. Our statistical system was able to identify all the three instances of this attack. The figure 4 refers to a particular instance of the "Mailbomb" attack occurred in the second day of the second week. As we see in figure 4 the attack occurred corresponding to the 390-394 time window. The peak that identifies the attack has a value much higher than the values shown for the background traffic.

3.2 Results of the Neural Subsystem

As we said, the two-tier neural architecture consists of SOM based on an unsupervised learning. The global

Table 1: Results related to the background traffic.

Class	Background traffic
0	4%
1	92%
2	1%
3	3%

architecture has to classify in a particular class a fragment of the network traffic. We hope that all the normal traffic is classified in the same class, while the attacks will be classified in classes different from the background traffic. The best case is that different instances of the same attack are classified in the same and reserved class, in a way that some classes are used to classify specific attacks. In other words the system would distinguish not only the background traffic from the traffic with attacks, but it can also detect the typology of attack. The four networks of the first layer present a two-dimensional lattice of neurons (5x5 dimension), so each network has 25 different classes in which to classify the input pattern. The network of the second layer is a one-dimensional chain, with 9 nodes, which would distinguish 8 different attacks and the background traffic. An important parameter to consider is the number of epochs used during the learning, that is the times in which the learning set is presented to the network. The experimental results demonstrated that considering an only one epoch the results concerning the classification of the input pattern were rather unsatisfactory. The results are better than the previous one using 10 epochs of learning. For each different epoch, the days used for the learning are presented according to a random sequence. For the training we considered the first, the second, the third and the fifth week. While for the testing phase we performed two different experiments. In the first example we tested the system with the only background traffic (considering the first and the third week only). While in the second example, we tested the system considering all the dataset (including background traffic and attacks).

The table 1 contains the results concerning the first experiment with the only background traffic. The neural subsystem mainly classifies the normal traffic in 1 class (with a rate of 92%). As shown in table 1, a very little rate of traffic has been classified in 0, 2 and 3 classes, this because, in our opinion, the behaviour of these fragments of traffic differs a little from a normal behaviour. Besides, some classes (from 4 to 8) are not used obtaining a classification rate equals to zero. So we expect that these classes with zero value, in the following will be reserved for the attacks. This aspect can be put in evidence considering the second experiment in which for the testing phase we considered all the dataset. The results of the second experiment are showed in table 2[1]. As table 2 shows, in some cases the ”Decisional Motor” recognized the particular attack classifying in an only class all the instances of the same attack. In fact, the three different instances of Mailbomb attack have been classified in the 8 class, the two instances of UdpStorm attack have been classified in the 4 class and finally two of three instances of the Smurf attack have been classified in the 3 class. So, the system reserved some classes for the classification of the same attack. This behaviour is very important for a system that detects intrusions since in this way it is able to recognize which typology of attack occurred. Concerning other attacks (Sshprocesstable, Mscan and Secret), the system recognized them as attacks, but it has not been able to distinguish them. In fact, it classified all these attacks in the 3 class.

Table 2: Results of the test related to the all dataset inclusive of attacks.

	Attacks (%)							
Cl	Mb	Sf	P	U	Ss	N	Mn	St
0	0%	0%	0%	0%	0%	0%	0%	0%
1	0%	0%	0%	0%	0%	0%	0%	0%
2	0%	0%	50%	0%	0%	0%	0%	0%
3	0%	67%	0%	0%	100%	0%	100%	100%
4	0%	0%	0%	100%	0%	0%	0%	0%
5	0%	33%	0%	0%	0%	0%	0%	0%
6	0%	0%	0%	0%	0%	0%	0%	0%
7	0%	0%	0%	0%	0%	100%	0%	0%
8	100%	0%	50%	0%	0%	0%	0%	0%

4 CONCLUSION

The aim of this paper is to propose a novel anomaly based IDS. The system consisted of two subsystems: a statistical subsystem (”Discriminator”) and a neural networks based subsystem (Decisional Motor). The first allows a preliminary visual distinction between a normal behaviour and an anomalous one. The neural subsystem has to establish the effective status of the network: if there is a normal situation or if an attack occurs. In order to analyze the network traffic from and towards the monitored host we considered 8 features: 4 for the input packets and the same 4 for the output packets (IP Address, number of Port, IP:Port and the number of packets). Analyzing the trend of these features we can evaluate the behaviour of the

[1] (Mb=Mailbomb, Sf=Smurf, P=Portsweep, U=UdpStorm, Ss=Sshprocesstable, N=Neptune, Mn=Mscan, St=Secret).

monitored server. The 8 features were preliminary processed by the statistical subsystem, and then the statistical results were classified by the neural architecture basing on their trend.

For the experimental phase, we used the most complete and available benchmark in Internet: the 1999 DARPA dataset. During the first experiment the system was able to quite correctly classify the background traffic. In fact, considering the only background traffic, a high rate (92%) of this traffic was classified in class 1. Also in the second experiment, considering all the dataset, we obtained encouraging results. The system was able to autonomously distinguish the normal traffic from the anomalous one: it classified the attacks in different classes from that used for the background traffic. A very important aspect is that in some cases the system recognized an attack classifying all the instances of the attack in the same class. For example, in the case of three different Mailbomb instances, for two UdpStorm instances, and finally for two of three Smurf instances. Besides, the system reserved a particular class to classify one typology of attack. In other words, a class is used to classify only different instances of the same attack. This means that our IDS is not only able to distinguish the normal traffic from the malicious traffic, but it can establish which attack occurs. Our evaluation is based on a single source of network traffic due to the lack of other available data. Obviously, every environment is different, so we plan to confirm our results using other sources of real traffic.

REFERENCES

Cabrera, J.B.D., Bavichandran, B., Mehra, R.K., 2000. Statistical Traffic Modeling for Network Intrusion Detection. *Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication systems*:466-473.

DeLooze, L.L., 2006. Attack Characterization and Intrusion Detection using an Ensemble of Self-Organizing Maps. *Proceedings of International Joint Conference on Neural Networks*, Vancouver (Canada):2121-2128.

Depren, O., Topallar, M., Anarim, E., Ciliz, M.K., 2005. An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks. *Expert System with Applications*, 29:713-722.

Ghosh, A.K., Michael, C., Schatz, M., 2000. A Real-Time Intrusion Detection System Based on Learning Program Behavior. *Proceedings of the 3rd International Symposium on Recent Advances in Intrusion Detection*:93-109.

Ghosh, A.K., Wanken, J., Charron, F., 1998. Detection Anomalous and Unknown Intrusions Against Programs. *Proceedings of IEEE 14th Annual Computer Security Applications Conference*:259-267.

Haines, J.W., Lippmann, R.P., Fried, D.J., Tran, E., Boswell, S., Zissman, M.A., 1999. 1999 DARPA Intrusion Detection System Evaluation: Design and Procedures. *MIT Lincoln Laboratory Technical Report*.

Kohonen, T., 2001. *Self-Organizing Maps*. 3rd edition, Springer-Verlag, Berlino.

Labib, K., Vemuri, V.R., 2004. Detecting and Visualizing Denial-of-Service And Network Probe Attacks Using Principal Component Analysis. *SAR'04 the 3rd Conference on Security and Network Architectures*.

Lee, W., Stolfo, S.J., Mok, K.,1999. A Data Mining Framework for Building Intrusion Detection Models. *Proceedings of 1999 IEEE Symposium of Security and Privacy*:120-132.

Mahoney, M.V., Chan, P.K., 2003. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. *Proceeding of Recent Advances in Intrusion Detection (RAID)-2003* LNCS 2820:220-237.

Valdes, A., Anderson, D., 1995. Statistical Methods for Computer Usage Anomaly Detection Using NIDES. *Technical Report*, *SRI International*.

Vigna, G., Kemmerer, R.A., 1998. NetSTAT a network-based Intrusion Detection Approach. *Proceedings of 14th Annual Computer Security Applications Conference*:25-34.

Ye, N., Emran, S.M., Chen, Q., Vilbert, S., 2002. Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection. *IEEE Transactions on computers*, 51(7):810-820.

Zhang, Z., Li, J., Manikopoulos, C.N., Jorgenson, J., Ucles, J., 2001. Neural Networks in Statistical Anomaly Intrusion Detection. *Neural Network Word, International Journal of Non-Standard Computing and Artificial Intelligence*, 11(3):305-316