Traffic Pattern Analysis for Distributed Anomaly Detection

Grzegorz Kolaczek and Krzysztof Juszczyszyn

Institute of Informatics,
Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{grzegorz.kolaczek,krzysztof.juszczyszyn}@pwr.wroc.pl

Abstract. Network anomalies refer to situations when observed network traffic deviate from normal network behaviour. In this paper, we propose a general framework which assumes the use of many different attack detection methods and show a way to integrate their results. We checked our approach by the use of network topology analysis methods applied to communication graphs. Based on this evaluation, we have proposed a measure called the AttackScore, which assesses the risk of an on-going attack and distinguishes between the effectiveness of the analytic measures used to detect it.

Keywords: Service Oriented Architecture, Security, Anomaly Detection.

1 Introduction

The most intensively explored approach to unknown threats detection is anomaly detection. Anomaly detection can be described as an alarm for strange system behavior. The concept stems from a paper fundamental to the field of security - An Intrusion Detection Model, by Dorothy Denning [4]. The aim of the anomaly detection is discovering of all abnormal states of the system in relation to the network traffic, users activity and system configuration that may indicate violation of security policy [8]. The general idea of protecting computer systems security with anomaly detection mechanisms is very simple, however implementation of such systems has to deal with a lot of practical and theoretical problems. The security assessment of a network system requires applying complex and flexible mechanisms for monitoring values of system attributes, effective computational mechanisms for evaluating the states of system security and the algorithms of machine learning to detect new intrusions pattern scenarios and recognize new symptoms of security system breach [8]. There are three fundamental sets of attributes that are considered in anomaly detection: basic (packet data), content (payload) and traffic (statistics)[5,8].

2 Related Works

The earliest anomaly detection-based approach, proposed by Denning, employs statistics to construct a point of reference for system behavior. The training

R. Wyrzykowski et al. (Eds.): PPAM 2011, Part II, LNCS 7204, pp. 648–657, 2012.
© Springer-Verlag Berlin Heidelberg 2012

of an anomaly detection sensor is accomplished by observing specific events in the monitoring environment such as system calls, network traffic or application usage, over a designated time period [10]. In many situations one would require constant training of detection system. The example of statistical anomaly detection is e.g. Haystack [11], Intrusion Detection Expert System (IDES)[12] Next-Generation Intrusion Detection Expert System (NIDES) [13]. Research done by Kruegel et al. [14] presents approach to find the description of a system using a payload byte distribution and extracted packet header features. The key aspect of this work is that we provide a quantitative evaluation of different approaches in a single evaluation framework to improve anomaly detection by parallel processing. Some previous work [15] demonstrate that combining multiple detection algorithms does offer an increase in performance over individual detectors. Besides, Gao et al. [3] proposed using an combination of different detection algorithms to build a more accurate model for continuously arriving data and proved theoretical improvement over each single algorithm. However, their work did not consider how to pick the best combination of algorithms. This paper propose distributed traffic pattern analysis to improve the anomaly detection in high speed networks where large quantities of network packets are exchanged by hundreds and thousands of network nodes. The evaluation of detection methods has been performed using a simulation of the Internet Worm attack traces. Compared with earlier works, the presented proposition is more specific in defining the traffic anomaly models.

3 Experiential Evaluation of Distributed Anomaly Detection

Network traffic show some quantitative and topological features that appear to be invariant and characteristic for given network. These distinct features concern topology of network communication, considered as origin-destination flows graph, the distribution of data volumes and the in/out ratio of data sent between nodes [9]. There is also a detectable dependence between worm propagation algorithm, and communication pattern disturbance[8]. Network traffic can be observed and analyzed according to several characteristic values such as: number of bytes send/received per second, number of packets, number of IP destinations, average packet size, etc.. Changing value of these parameters may be viewed as an important source of information about network host of link state. This correlation between network traffic and security breaches has been used in several network intrusion detection systems. For example the following relations between security incident type and observed network traffic parameters change were observed by Anukool Lakhina et al.[1]:

Proposed distributed anomaly detection method will gather information about communication within the network. Then the existing communication patterns will be discovered. The system will be viewed as a graph consisting of nodes and edges which appear if there exists data flow between given pair of nodes. The observation of communication patterns allows to tune the system and track anomalies which are hard to detect on the basis of traffic observations alone.

Table 1. Relations between security incident type and observed network traffic parameters

Security Incident	Traffic anomaly observed
ALPHA	Single source and destination address are dominant with the noticeable number of bytes, number of packets values increase.
DOS,DDOS	Large increase of number of packet with the same (IP,port) pair in the destination address while the distribution of source addresses remains almost unchanged.
SCAN	Increase of flows with the same source address and various combinations of (IP,port) in destination address. Packets with the similar size are dominant.
WORM	Flows with one dominant port in destination address can be observed.

3.1 Modelling Internet Worm for Anomaly Detection Method Evaluation

Internet worms are programs that self-propagate across a network exploiting security or policy flaws in widely-used services [1]. The taxonomies of malware distinguish several types of Internet worms, but there are two major classes of them, scan-based worm and email worms which require some human interaction to propagate and thus propagate relatively slowly. Scan-based worms propagate by generating IP addresses to scan and compromise any vulnerable target computer . This type of worms could propagate much faster than email worms [2]. For example, Slammer in January 2003 infected more than 90% of vulnerable computers in the Internet within just 10 minutes [3]. The basis of our Internet worm modeling is the classical epidemic model [6]. The experimental test bed is assumed to be a homogeneous network any infectious host has the equal probability to infect any susceptible host in the system. Once a host is infected by a disease, it is assumed to remain in the infectious state forever. Two experiments been proposed to evaluate distributed anomaly detection method:

1. Sequential Scanning: This scenario lets each newly infected system choose a random address and, then scans sequentially from there.
2. Hit-list worm: A hit-list worm first scans and infects all vulnerable hosts on the hit-list, then randomly scans the entire network to infect others. It has been assumed that the hit-list comprises the well known address to an infected host.

Common assumptions for all experiments performed are:

- N= 1000 - The total number of host hosts in experimental network
- V=30 - The population of vulnerable hosts in experimental network. It has been assumed that the number of vulnerable hosts is approximately 3% of all population [5]

– I=100 - Average scan rate. The average number of scans an infected host sends out per unit time (time window).

The normal communication activity for experimental network has been modelled using Barabasi scale-free network model [7] with $\gamma = 3$. It was also assumed that the communication is being observed in consecutive time windows with some perturbations during normal network operation which reflect the everyday variance of communication. We have generated realistic communication patterns which join the variance with the properties of a scale-free network. The worm related communication patterns has been added to these normal patterns according to the abovementioned Sequential-scanning and Hit-list scenarios.

3.2 Evaluated Anomaly Detection Algorithms

The idea behind our approach to traffic anomaly detection was to apply structural network analysis in order to compare the topology of communication network during normal operation and during an ongoing attack. We have applied the analysis of role-set structure of a network based on the similarity of link profiles among its nodes. In general structural equivalence measures may be divided into three groups:

– Match measures assuming matching between all pairs of node profiles, usually based on set similarity measures like Jaccard Coefficient etc.
– Correlation measures based on correlation measures applied to node profiles (which are treated as vectors): Cosine, Pearson, Spearman.
– Distance measures measuring the distance between points in n-dimensional space which represent node profiles.

For our experiments we have chosen seven structural equivalence measures:

– Match measures: Jaccard, Phi, Braun and Blanque
– Correlation measures: Pearson, Inner Product
– Distance measures: Euclidean, Bhattacharyya Distance

The interpretation of the results returned by the match and correlation measures is that they are similarity metrics. From the other hand, the distance metrics are the opposite bigger distance stands for more dissimilarity. From this point on, we will refer to all the measures as similarity measures as in fact- we use them to assess how the actual structure of communication network differs from the one emerging from normal system operation. To allow the comparison between the used measures all results were normalized.

3.3 Structural Equivalence Measures during Normal Network Operation

First step in our analysis was to assess the performance of our similarity measures under assumption that there is no attack, and the changes in the communication

Table 2. Structural similarity between communication networks during normal operation

Time step	Jaccard	Phi	Braun, Blanque	Pearson	Inner Product	Euclidean	Bhatt. Distance
1	0,851	0,875	0,992	0,903	0,717	0,770	0,670
2	0,910	0,923	0,998	0,919	0,787	0,802	0,679
3	1,000	1,000	0,990	1,000	1,000	0,864	0,702
4	0,948	0,954	1,000	0,964	0,871	0,813	0,681
5	0,847	0,872	0,987	0,869	0,688	0,782	0,676
Mean value	0,911	0,925	0,998	0,931	0,813	0,806	0,682
Std deviation	0,058	0,048	0,005	0,046	0,113	0,032	0,011

network structure reflect normal operation. The simulation was carried on for six consecutive time windows. For each of these windows the communication network with links reflecting message exchanges between nodes) was created. Table 1 presents the structural similarity between the first and the consecutive time windows as assessed by seven similarity and distance measures.

We assume that the similarity and distance values around the mean value reflect normal network operation. We define an attack as a situation when the similarity differs from the mean value computed on the basis of history more than doubled standard deviation (this attack threshold may be of course tuned in the case of real systems in order to reflect the changes in given network during normal operation). This restrictive assumption may eventually lead to the false attack detection, in the case of data taken from Table 1 this is Bhattacharyya Distance in step 4 or Braun and Braun and Blanque in step 6. In order to avoid false alarms caused any of the measures, we assume that an attack must be confirmed by at least two of them.

3.4 Anomaly Detection

In the second step of our experiments we have checked the influence of Hit-list and Sequential Scanning attacks on the network topology. Fig. 1 presents the results obtained for the first 5 time steps of an ongoing sequential attack.

An immediate consequence of the first infections is the scanning procedure performed by the infected nodes which inevitably leads to the emergence of hubs in communication network which disturbs the network structure and results in visible changes in similarity and distance measures. We can see the growing difference between attacked communication networks and the normal patterns of communication recorded prior to the attack. The only exception is the Inner Product measure, which seems not to distinguish between normal and attacked networks. Note, that distance measures have growing values for older phases of the attack, while match and correlation measures (interpreted as similarity) are decreasing. The same is visible on Fig. 2 which shows similar results for a hit-list

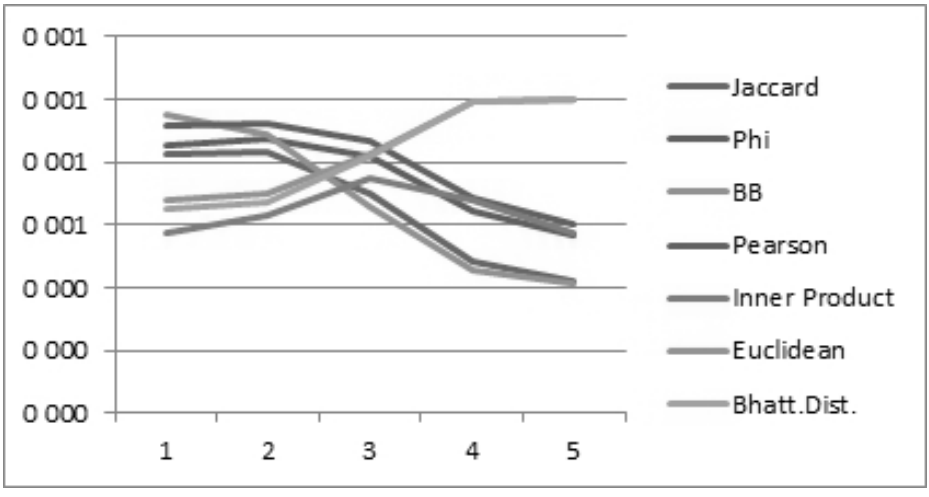

Fig. 1. Similarity and distance measures between normal communication network and the network under sequential attack

Table 3. Attack detection (time steps)

	Jaccard	**Phi**	**Braun, Blanque**	**Pearson**	**Inner Product**	**Euclidean**	**Bhatt. Distance**
Seq. scanning detected in step:	4	4	2	5	6	5	4
Hit-list detected in step:	3	5	2	5	6	5	4

attack. Despite similarities we can also see the differences between Fig. 1 and Fig. 2 they reflect the fact that both modeled attacks have different dynamics, the hit-list worms use the local address lists at first. In result, the number of infected nodes is not so rapid as in the case of hit-list, which is reflected by the moderate (when compared to Fig. 1) change of our measures.

The results presented on the Fig. 1 and Fig.2 may be confronted with the values presented in Table 1, which allows to determine the time step in which each of the measures will report an attack (Table 3).

3.5 Algorithm Aggregation

From Table 3 we can notice that our measures have different performance for each of the considered attacks, there are also differences in the case of attack detection during sequential scanning and hit-list attacks. We define similarity

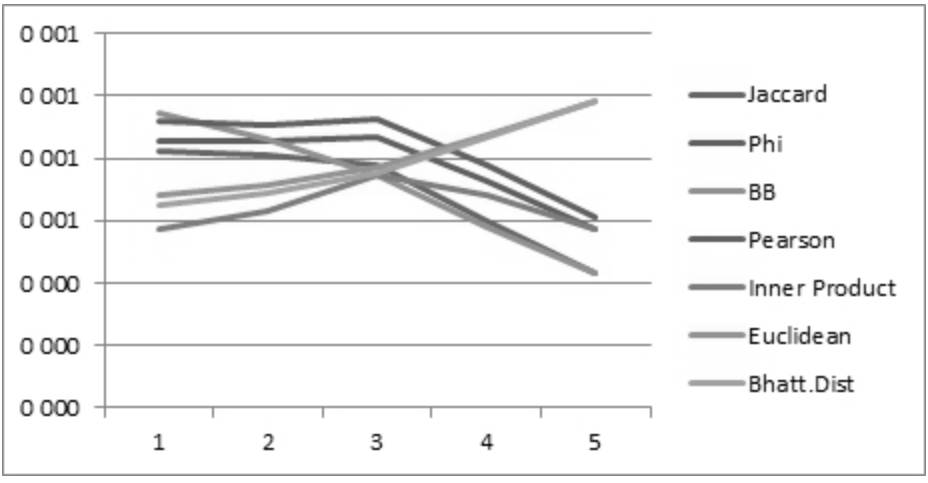

Fig. 2. Similarity and distance measures between normal communication network and the network under hit-list attack

between our measures in terms of their decisions about attack detection. For any two measures m_1 and $m_2$their similarity is defined by comparing the total number of their decisions and the number of decisions in which they have agreed: A_{00} and A_{11} are the number of cases where both measures decided that there is respectively no attack and attack, while A_{01} and A_{10} are the numbers of cases in which they did not agree.

$$SIM(m_1, m_2) = \frac{A_{00} + A_{11}}{A_{00} + A_{11} + A_{01} + A_{10}} \tag{1}$$

Table 4 shows the results for all pairs of measures on the basis of our experiments (the similarity matrix with respect to the decisions of the measures is symmetrical).

Single attack alert (raised by only one measure) may be caused by the normal fluctuations occurring during normal network operation and should not be treated as security breach. We assume that we must get confirmation by at least two of the measures. However, (Tab.4) some of them show close similarity of their results, which fact should be taken into account. In our approach we use a form of "weighted voting" which leads to generation of joint opinion of the measures about the attack. The following rules are applied: 1. At least two measures must positively recognize the attack. 2. When condition 1. is fulfilled a special measure, called Attack Score (AS) is applied to all the measures which raise an alarm.

$$AS = \frac{1}{n_{A^2}} \sum_{j=1}^{n_A} \sum_{i=1}^{n_A} \left(1 - Sim\left(m_i^{attack}, m_j^{attack}\right)\right) \tag{2}$$

Table 4. Similarity between attack detection measures

	Jaccard	**Phi**	**Braun, Blanque**	**Pearson**	**Inner Product**	**Euclidean**	**Bhatt. Distance**
Jaccard	1,000	0,833	0,750	0,750	0,583	0,750	0,917
Phi		1,000	0,583	0,917	0,750	0,917	0,917
Braun, Blanque			1,000	0,500	0,333	0,500	0,667
Pearson				1,000	0,833	1,000	0,833
Inner Product					1,000	0,833	0,667
Euclidean						1,000	0,833
Bhatt. Distance							1,000

Table 5. Attack detection (time steps)

Time window:	**1**	**2**	**3**	**4**	**5**	**6**
Seq. scanning	0	0	0	0,111	0,170	0,217
Hit-list	0	0	0,125	0,148	0,170	0,217

In the above equation n_A is the total number of measures which confirm the attack (lets call them m_1^{attack}, m_2^{attack}, $m_{n_A}^{attack}$). Thus, AS computes the sum of similarities for all possible pairs of attack-reporting measures complemented to 1, then returns their average value. Self-similarity of the measures is zeroed. In this way, if the similarity of the scored measures is high, SA will be significantly lower, then in the case they are behaving in a different way. In result SA promotes the attack reports confirmed by a measures which show different behavior.

The AS reaches its highest value, when all the measures agree about the attack (for our experiment it was 0,217). However it promotes the results returned by the measures which differ from each other in the context of normal network communication. This can be seen in the case of the fourth time window where AS is lower for (Jaccard, Phi, Bhattacharyya Distance) in Seq.Scanning then it is for (Jaccard, Braun-Blanque, Bhattacharyya Distance) in HitList case. This is because the higher difference between the measures recognizing the HitList attack. The higher attack score reflects that it is recognized by the measures which use not the same definition of the structural connection pattern of the network. Moreover, our framework is general and may be applied in the case of measures which differ from each other according to algorithms, nature and the grounding data.

4 Conclusions and Future Work

We have presented an original approach which allows to us different measures for the detection of abnormal network communication patterns. It was tested

on a large network which reflects the scale-free pattern and statistics of the networks detected in different forms of communication . We have also proposed the application of graph structural equivalence measures to the detection of attacks and tested it on the simulated attack occurring in the sample network. Our framework will be further developed in the following directions:

- Detecting the attack type: from the figs 1 and 2 we may notice that different type of attack result in different behavior of our detection. In result the attacks led by various algorithms may be distinguished from each other.
- Instead of simple structural network measures used in our test case, the other sophisticated methods may be applied. Our framework is flexible enough to accommodate and reflect the differences between the measures used.
- SA definition as a weighted voting approach leaves space for checking the interplay between the attack threshold level and the effectiveness of the method.

The first application for our approach will be a SOA system providing educational and administrative services at the Wrocaw University of Technology. The software agents collecting data about normal communication patterns in the system will be developed [15].

Acknowledgements. The research presented in this work has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

References

1. Asokan, N., Niemi, V., Nyberg, K.: Man-in-the-middle in tunnelled authentication protocols. Technical Report 2002/163, IACR ePrint archive (2002)
2. Balasubramaniyan, J.S., Garcia-Fernandez, J.O., Isacoff, D., Spafford, E., Zamboni, D.: An Architecture for Intrusion Detection Using Autonomous Agents. In: Proceedings of the 14th Annual Computer Security Applications Conference (1998)
3. Li, P., Gao, D., Reiter, M.K.: Automatically Adapting a Trained Anomaly Detector to Software Patches. In: Balzarotti, D. (ed.) RAID 2009. LNCS, vol. 5758, pp. 142–160. Springer, Heidelberg (2009)
4. Denning, D.E., Edwards, D.L., Jagannathan, R., Lunt, T.F., Neumann, P.G.: A prototype IDES: A real-time intrusiondetection expert system. Technical report, Computer Science Laboratory, SRI International, Menlo Park (1987)
5. Kolaczek, G., Pieczynska-Kuchtiak, A., Juszczyszyn, K., Grzech, A., Katarzyniak, R.P., Nguyen, N.T.: A Mobile Agent Approach to Intrusion Detection in Network Systems. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 514–519. Springer, Heidelberg (2005)
6. Onnela, J.P., Saramaki, J., Szabo, G., Lazer, D., Kaski, K., Kertesz, J., Barabasi, Hyvönen, A.L.: Structure and tie strengths in mobile communication networks. Proceedings of the National Academy of Sciences 18, 7332–7336 (2007)
7. Park, J., Barabási, A.L.: Distribution of node characteristics in complex networks. Proceedings of the National Academy of Sciences of the United States of America 104(46), 17916–17920 (2007)

8. Patcha, A., Park, J.-M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12), 3448–3470 (2007)
9. Scott, J.: Social Network Analysis: A Handbook, 2nd edn. Sage, London (2000)
10. Anderson, D., Lunt, T.F., Javitz, H., Tamaru, A., Valdes, A.: Detecting Unusual Program Behavior Using the Statistical Component of the Next-generation Intrusion Detection Expert System (NIDES), Computer Science Laboratory, SRI International, Menlo Park, CA, USA SRI-CSL-95-06 (May 1995)
11. Smaha, S.E.: Haystack: An intrusion detection system. In: Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, pp. 37–44 (1988)
12. Lunt, T.F., Tamaru, A., Gilham, F., Jagannathm, R., Jalali, C., Neumann, P.G., Javitz, H.S., Valdes, A., Garvey, T.D.: A Real-time Intrusion Detection Expert System (IDES), Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Final Technical Report (February 1992)
13. Kruegel, C., Mutz, D., Robertson, W., Valeur, F.: Bayesian event classification for intrusion detection. In: Proceedings of the 19th Annual Computer Security Applications Conference, Las Vegas, NV (2003)
14. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for unix processes. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA, pp. 120–128 (1996)
15. Kołaczek, G.: Multiagent Security Evaluation Framework for Service Oriented Architecture Systems. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009. LNCS (LNAI), vol. 5711, pp. 30–37. Springer, Heidelberg (2009)