NADA – Network Anomaly Detection Algorithm

Sílvia Farraposo[1], Philippe Owezarski[2], Edmundo Monteiro[3]

[1] School of Technology and Management of Leiria
Alto-Vieiro, Morro do Lena, 2411-901 Leiria, Apartado 4163, Portugal
[2] LAAS – CNRS, 7 Avenue du Colonel Roche
31077 Toulouse, CEDEX 4, France
[3] Laboratory of Communications and Telematics, Computer Science Department
Pólo II – Pinhal de Marrocos, 3030-290 Coimbra, Portugal

silvia@estg.ipleiria.pt, owe@laas.fr, edmundo@dei.uc.pt

Abstract. This paper deals with a new iterative Network Anomaly Detection Algorithm – NADA, which is threefold: it accomplishes the detection, classification and identification of traffic anomalies. Our approach goes one step further than others since it fully provides all information required to limit the extent of anomalies by locating them in traffic traces, identifying their classes (e.g., if it is a Denial of Service, a Network Scan, or other type of anomalies), and giving their features as, for instance, the source and destination addresses and ports being involved. For this purpose, NADA uses a generic multi-featured approach executed at different time scales and at different levels of IP aggregation. Besides that, the NADA approach contributed to the definition of a set of traffic anomaly signatures. The use of these signatures makes NADA suitable and efficient to use in a monitoring environment. NADA has been validated using data traces containing documented anomalies as the one gathered in the MetroSec project.

Keywords: Traffic anomaly Identification, Anomaly Signature.

1 Introduction

The lack of security in networks is an issue that network administrators would like to solve on the fly, independently of the network size. Anomalies being a structural part of traffic, it is at least important to completely detect, classify and identify them in order to act adequately. The computing of the detected anomalies will be different if they are legitimate, as a flash crowd, or illegitimate, as a DoS attack.

The diversity of network dysfunctions motivated the development of NADA that stands for Network Anomaly Detection Algorithm. Its objectives are threefold:

- Detection of anomalies, i.e. determining whether an anomaly is occurring;
- Classification, i.e. determining what kind of anomaly arises. This means determining whether the anomaly is legitimate or illegitimate, and its precise kind (HTTP flash crowd, scanning, SYN flooding, etc.);

- Identification, i.e. being able to determine all the anomaly constituting packets and flows.

In addition, NADA aims at being completely generic, and can work on any kind of time series issued from incoming traffic or packet traces. To illustrate NADA, in this paper we will consider three different data time series:

- Number of packets per unit of time;
- Number of bytes per unit of time;
- Number of new flows per unit of time.

We insist here on the need of working on several time series to allow the correct detection, classification and identification of anomalies, since each type of anomaly acts differently over each of the parameters presented above. In addition, as we want the tool based on NADA, currently under development, to be efficiently used by network operators, it has to work on network and traffic representative features, as bytes, packets and flows, and simple statistics. Times series about the number of SYN or RST packets could also be easily added in the algorithm if they are meaningful for the operators. In any case, by using only simple mathematics, we want to make NADA easily and efficiently exploitable and configurable by any network technicians.

The detection capability of NADA by screening a trace allows spotting time intervals where a significant traffic variation has occurred. The classification capability permits among a set of possible traffic anomalies to signal which is occurring. This point is particularly important, since a variation previously signaled may not correspond to a traffic anomaly, but only to a normal traffic variation. The information required to perform the classification of anomalies is produced by NADA through the simultaneous analysis of the different data time series, at different time scales, and different IP levels of aggregation. Finally, the identification feature of NADA gives information about the entities involved in the anomaly, source and destination addresses and ports. Also, it was noticed that the distribution of the IP features characterizing a traffic anomaly has a graphical pattern that is easily and uniquely identifiable. Such patterns can be useful when using NADA in a monitoring environment.

Other approaches for detecting traffic anomalies exist. However, as far as we know, none permits simultaneously the detection, classification and identification of traffic anomalies. Most of the earlier work in this subject was mainly concerned with the detection of anomalies as the ones of Barford et al. [1] and Krishnamurthy et al. [2]. Or was mainly anomaly-oriented, as the work of Hussain et al. [3] that proposed a taxonomy for DoS attacks, the work of Jung et al. [4] that studied flash crowds, or the system developed by Guo et al. [5] that detected and protected networks from network scans, to name a few. One major improvement in the anomaly detection field was the introduction of some level of classification in the algorithms being proposed, using information provided by IP features. Important contributions were the ones of Kim et al. [6], Estan et al. [7] and Lakhina et al. [8]. Nevertheless, anomaly classification and identification remains an important, unmet challenge, since none of the proposals exploited exhaustively the richness of IP feature information to provide accurate information about the actors of a traffic anomaly by identifying them. This is in particular the fault of the complex mathematics methods which make difficult to come back from the frequency or entropy spaces to network features easily

understandable by network technicians operating networks. Our work, through a correct collection of traffic time series permits not only the detection and classification of different kinds of anomalies, but also to identify the intervening parties, in an easy way for both configuring the tool and analyzing its outputs. This aspect is particularly important when one of the main goals is to limit the negative effects of an anomaly occurrence in real networks.

The rest of this paper is organized as follows: Section 2 gives an overview of the NADA algorithm presenting its main features. Section 3 presents the traffic database that was used to assess NADA. Section 4, through intensive use of NADA on the traces of our database, presents anomaly signatures, and how these signatures can be used for anomaly classification. Section 5 presents an assessment of NADA efficiency, and section 6 concludes the paper, summarizing our ongoing research.

2. Network Anomaly Detection Algorithm – NADA

NADA has been defined as a multi-scale (there is not a unique time-scale to detect anomalies), multi-criteria (anomalies do not affect traffic criterions in the same way), and multi-level of IP aggregation (each type of anomaly is better perceived at different levels of flow aggregation) approach. For illustrating one case of the capabilities of NADA, in this paper, the screening of anomalies is done at the Internet Layer of TCP/IP stack, using packet or flow information, according to the Claffy et al. flow definition [9]. NADA is an approach with two phases. The first one is mainly devoted to the detection and classification of traffic anomalies, while the second stage targets the anomalous flows by fully characterizing them.

2.1 Detection and Classification of Anomalies

The first phase of NADA is a recursive process in which the lower stages consist of low cost anomaly detection mechanisms that provide information about the existence or not of an anomaly and permits the reduction of search space for further traffic analysis. Successive stages operate on data streams of progressively increasing granularity (e.g., packet information), and perform more fine-grained analysis.

The detection basis used in NADA is that any anomaly will be responsible for some variation at least on one of the criterions considered, at some time-scale and at some level of IP aggregation. The different time-scales chosen to screen traffic traces can range from very low values as some microseconds to several hours. The values used are a function of the traffic trace duration, but also of the type of anomalies we are looking for (some minor anomalies with respect to their impact on the network, are more easily detected when considering smaller time windows). The values for the IP aggregation level, can range from /0 (when all IP packets are considered) to /32 (when only the packets from or to a specific address are considered). Variations are pointed by using the formula below (1), in which X is a data time series directly obtained from traffic traces, and P is a data series that is obtained from X, and in which each value is the difference between two consecutive values of X.

Considering P instead of X is important because our formula considers that significant anomalies are not those that produce a variation in traffic flow, but those that are responsible for unexpected significant variations, that might disturb network resources. These significant variations were named deltoids by Cormode et al. in [10] who used them to detect significant traffic changes.

$$X = \{x_1, x_2, ..., x_n\}, x_i = \{\#\ packets | \#bytes | \#\ flows\} / \Delta \quad (1)$$
$$P = \{p_1, p_2, ..., p_{n-1}\}, p_i = x_{i+1} - x_i$$
$$\begin{cases} pi \geq E(p) + k\sigma, select \\ pi < E(p) + k\sigma, reject \end{cases}$$

The mean and the standard deviation, E(p) and σ respectively, of each time series are calculated and used to define a threshold. Each time slot for which one of its elements exceeds this threshold is a time slot in which a significant variation has occurred. This sort of filtering can be more or less coarse grained depending on the value of the adjustment parameter K of the formula, where smaller values of K fine-grain the search.

For each process stage a new set of time series is calculated, each being more specific than its predecessor. At the beginning, the time series are obtained considering all the packets of the trace, or a level of aggregation /0. At the second iteration time series are calculated for /1 flows, at the next iteration /2, and so on until /32. To reduce the execution time, and because of redundancy between two consecutive levels of aggregation, only some levels of aggregation are considered during the analysis (e.g. /8, /16, /24, /32, and some intermediate values if needed during the different algorithm iterations).

The result of this phase is a collection of flows, located in time that experience some significant variations, and because of that are possibly anomalous. The classification stage is based on signatures profile. It was noticed from previous executions of NADA over our trace database, that anomalies signaled by our algorithm presented always a set of characteristics that could be used to identify them in a one-to-one way. These anomaly signatures are a function of the entries presented in Table 1.

Table 1. Parameters used in anomaly signatures definition.

Parameter	Description
Time hierarchy	Time windows at which the anomaly is detected. Could range from one microsecond to several hours.
Network hierarchy	For source and target. Networks can be considered at different aggregates, e.g., /0 (the entire IPv4 address space); /8; /16 or /24 subnets; or /32 (single IP address).
Data time series	Data time series is a general term to identify any type of time series being considered. At this point the number of packets, bytes and new flows that presented a significant variation.
K	Level of filtering.

2.2 Identification of Anomalies

The purpose of anomaly identification is to allow its complete mitigation. This step then includes an exhaustive description of the anomaly, using all the information previously collected. Particularly, in this paper anomaly description is accomplished at the IP level, but could also be at a different point of view level. So, by executing our identification module and using the information obtained from the previous step it is possible to provide the features indicated by Table 2 about any anomaly being detected.

Table 2. Parameters used in anomaly identification.

Parameter	Description
Time window	Time window at which the anomaly is detected.
Source identification	Collection of IP addresses and ports responsible for the anomaly.
Target identification	Destination IP address and ports which receive suspect packets

3. Traces for NADA Assessment

To assess the accuracy and performance of NADA, a set of traces with well-defined and documented anomalies was required. Because such repository of traces does not exist, one was created in the framework of the MetroSec Project [11], between 2004 and 2006. This repository spans different types of anomalies, legitimate and illegitimate ones, and with different levels of intensity. Also, different types of anomaly generators were used in order to improve the quality of the database. Currently, our database has 42 entries distributed as showed in Table 3.

Table 3. The MetroSec database of anomalies: summary of traces.

Anomaly	Quantity	Tool	Intensity
Flash Crowd	4	Real browsing of web server	34% - 71%
DDoS	4	Hping	28% - 99%
	10	Iperf	15% - 58%
	3	Trinoo	7% - 87%
	9	TFN2K	4% - 92%
	12	TFN2K Modified	1% - 4%

As it can be seen from Table 3 the contribution of the attacks to the global throughput of the monitored link is highly variable (the intensity column), depending on the attack parameters and ranging from a major impact on global traffic profile to attacks that are completely hidden in the global traffic. However, the majority of the conduced attacks have remained low in traffic volume so that they cannot be easily

detected via simple statistics such as sample mean or variance estimates. Low intensity attacks are also meaningful for NADA, since one of its main purposes is the detection of anomalies even and mostly when their intensity level remain low, i.e., before they have a negative impact on the network QoS.

Though artificial or simplistic this approach of creating a database of anomalies may seem we see this reference database production methodology as a mandatory step for reliable development and validation of NADA.

3.1 Distributed Denial of Service Anomalies

The Distributed Denial of Service (DDoS) attacks were conduced by using a flooding approach with different types of tools. So, flooding with UDP, TCP, ICMP and mixed packets were executed, and different tools were used: Iperf, Trinoo, TFN2k, and a modified version of TFN2k (such version sends bursts of flooding instead of continuous ones, each with a different intensity). The use of different tools made possible to evolve from a simple DDoS solution using Iperf and Hping which are not per si a DDoS tool, to more complex scenarios in which the type of packets, the throughput, the duration, the DoS flow intensity, packets length and sending rate, as well as other parameters might be configured. Independently of the tool adopted, the played scenario included 4 attacking sites that flooded one machine located at LAAS in Toulouse, where the traffic was collected. The traffic related to these attacks was sent over the French operational RENATER network.

In addition, the database contains a range of attacks with different intensity levels having different impact levels on global traffic profile. The intensity of attacks is ranging from ones increasing the amount of traffic by more than 50% which are easily detected by most of the existing tools, to other that just increase the amount of traffic by few percents, and that are completely hidden by the normal variability of regular traffic. This aspect is important for the validation of NADA since we want a tool that is able to detect low intensity anomalies on a link, which could be part of a large scale DDoS, made for badly impacting network QoS when all flows aggregate.

3.2 Flash Crowd Anomalies

The correct classification of anomalies as Flash Crowds (FC) remains an important matter since they are legitimate anomalies, and actions taken in presence of DoS attacks (as the removing of anomalous flows) are not a solution in these cases. FC are used as examples of increase of traffic that are not attacks.

Particularly, in the MetroSec context, FC over a web server were conduced. To accomplish that, and to assure an experiment as real as possible, FC were not generated using automated tools, but instead of that, with real volunteering humans. As in the case of DDoS anomalies, the target server was the LAAS's web server (http://www.laas.fr), which was browsed in an aggressive way during a certain amount of time. However, the way each participant browsed the server was left at its own consideration. Precise starting and stopping times were defined, which lead to

anomalies with a duration of approximately 30 minutes, captured in 2 hours and 30 minutes long traces.

4. Classification Based on Anomaly Signatures

In this section we are presenting some of the results obtained when applying NADA to our documented database, studying how it behaves in presence of the anomalies we artificially added to the normal operational Renater traffic. The successive utilization of NADA showed that anomalies of a specific type have a signature, independently of the tool used to generate them, the type of packets being used, or the attack intensity. This result is not surprising as, by construction, NADA has been designed for this purpose.

The temporal localization of a possible anomaly is an important step, since hereafter, NADA targets only the time windows where anomalies were spotted. This helps to decrease execution time. At that point, NADA starts its classification function by analyzing each flow on that time window. If a flow is recognized as a regular one, it will not be considered for the next iterative screening stage, for that particular window. At the end, if an anomaly exists on that time window and the considered time series, it is possible to define a unique anomaly signature that may be used by the classification process of NADA.

For example, when analyzing documented traces with DDoS attacks, as expected, we have obtained four different types of signatures, depending on the number of sources that were flooding the target, and how port numbers were used. To mimic real attacks, for some attacks the same source port number was always used, while in others, each source packet has a different port number. Let consider one of our traces that contains a TCP flood performed by TFN2K and that floods the target machine with an intensity of 12% (i.e., the ratio between the number of packets due to the attack and the total number of IP packets on the link during the attack). Because of the capacity of the LAAS access link, this attack is far from disturbing the quality of cross connections. This type of anomaly is however easily detected when analyzing for a destination under suspicion how sources send packets to it. Figure 1 depicts it at flow level, when a /24 level of IP addresses aggregation is considered.

The shape of the plot in Figure 1 is characteristic of packet flooding attacks to a single destination. We can see on Figure 1 that 3 different sources (the most significant ones, distributed per three different plans) are using different source and destination ports for each packet sent to flood the target. If the picture of Figure 1 gives a clue about the type of anomaly, it is completely corroborated by the four plots that constitute what we have called the graphical signature (Figures 2 and 3). Those plots clearly show, for each suspect flow, how the source/destination IP addresses and ports relate to each other. We have seen, that their shape is always the same for each type of anomaly, and different between each others. This signature is unique for a given type of attack when considering the right time series.

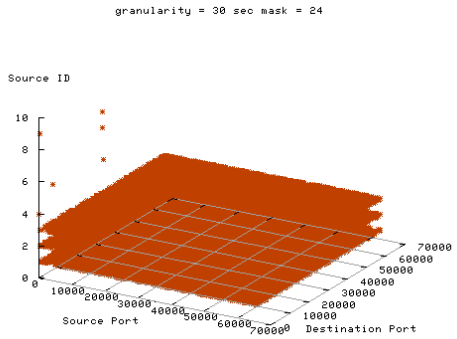
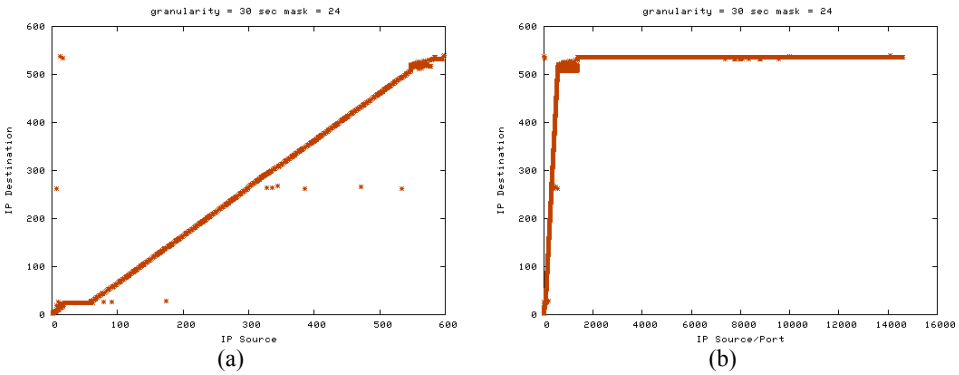


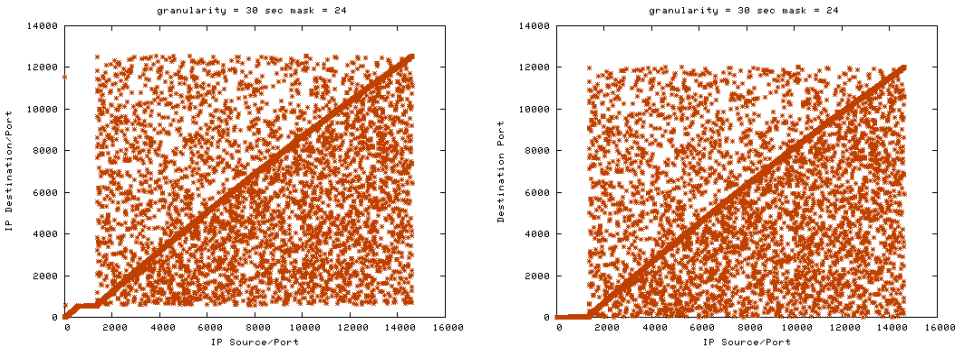
Fig. 1. Distribution of IP sources and source/destination ports for a destination IP address under suspicion.

For instance, for the anomaly being presented, we have obtained the four shapes depicted in Figure 2, where we can find the signature for a DDoS of type *n*S, *n*P : *1*D, *n*P (*n* IP sources using *n* Ports attack *1* IP destination using *n* Ports). Figure 2 shows how the different source and destination addresses/port relate to each other in flows under suspicion. So, at this point, only flows that might have some contribution to the anomaly are depicted. Recall that the plots presented in Figure 2 were obtained with a /24 aggregation level, meaning that more accurate results can still be obtained.

In Figure 2 (a) we can see that approximately at destination number 520 (that we call target), an aggregate of points denotes the different attacking sources. In Figure 2 (b), when adding the port number to each source, we obtain for the same target, a straight line denoting that each source is using different ports when flooding packets. One step further adding the port number to each destination, on figure 2 (c), we can see a straight line with a slope, showing that different ports of the target are flooded. Finally, on figure 2 (d), we can observe that the previous shape remains the same, meaning that the number of destination addresses attacked is very low when compared with the number of destination ports (this type of plot is particularly important to differentiate network scans from port scans).



(a) (b)



(c) (d)

Fig. 2. Visual signature for a DDoS attack obtained from a MetroSec trace. Plot (a) relates the different source IP addresses, with the destination IP addresses. Plot (b) relates de different source IP addresses/ports with the destination IP addresses. Plot (c) presents the source IP addresses/ports vs. the destination IP addresses/ports. Finally, plot (c) relates the source IP addresses/ports with the IP destination ports.

Figure 3 depicts the standard unique signature (without noise) for this kind of anomaly.

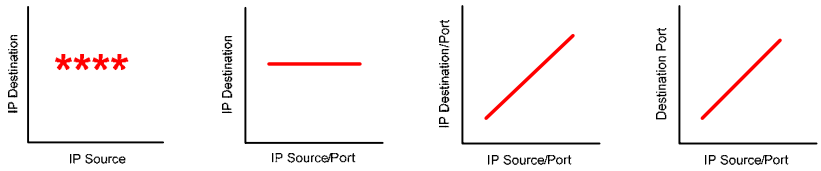


Fig. 3. DDoS standard signature without noise. Type of DDoS: *n* IP Sources, *n* Source Ports : *1* IP Destination, *n* Destination Ports.

It was noticed that besides the visual signature, the parameters used by NADA and described in Table 1, also changed according to the type of anomaly being detected. Hence, in the case of DDoS attacks it was observed that such type of attack is signaled at time scales lower than 60 seconds and when using levels of traffic aggregation higher than /16. Also, it was noticed that this type of anomaly is responsible for a significant variation in the number of packets and bytes, at all time scales and levels of aggregation, and that the value of K must be decreased when attacks are of low intensity (less than 10%).

Running NADA on the traces collected in the MetroSec project, permitted the isolation of other anomalies not due to artificial experiments. One of the most common attacks observed is network scan whose main purpose deals with searching for network's flaws. Figure 2, besides the DDoS signature, also contains the signature for a network scan of type *n*S, *n*P : *n*D, *1*P (*n* IP sources using *n* Ports scan *n* IP destinations using *1* Port). Removing all the noise, the signature is similar to the one depicted in Figure 4.

The sequence of patterns in Figure 4 is detected in the sequence of plots of Figure 2. From figure 2(a) we can observe that the network scan involves almost 500 IP sources and more or less the same number of IP destinations (slope depicted in figure). The same slope is still visible at plots 2(b) and (c), being reduced to straight line in the last plot, since a single destination port is being used with every anomalous packet (in this case it was port 139).

Beside the visual signature, network scans are signaled by a significant variation in the number of packets and bytes, only detected when considering smaller time scales (less than 30 seconds). However, and in opposition to DDoS anomalies, it was noticed that as traffic flow is disaggregated, the network scan vanishes, and it is no more detected. This usually occurs when considering an aggregation level higher than /24.

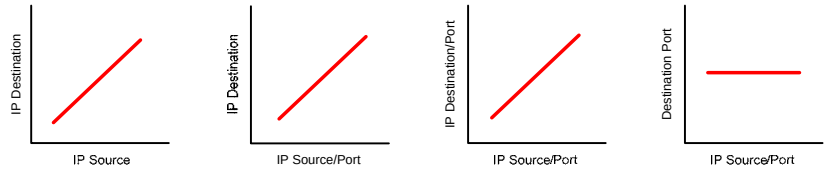


Fig. 4. Network scan standard signature without noise. Type of network scan: *n* IP Sources, *n* Source Ports : *n* IP Destination, *1* Destination Port.

5. NADA Assessment

One important aspect when developing an algorithm such as NADA is its effectiveness in the accomplishment of its objectives: detection, classification and identification of traffic anomalies.

From our experience we know that the ability of any algorithm of this type to accurately perform depends of several external factors besides the algorithm itself. Namely, the traces used to test the algorithm, the type of anomaly, the anomaly intensity and the anomaly duration. Particularly, NADA is an algorithm that requires the tuning of parameters to perform correctly. The filtering factor K is one of the most important in our point of view: smaller values correspond to less significant changes, and in the other hand, higher values of K permit only the detection of anomalies that have a significant impact in traffic. Our database of traffic traces containing anomalies has been especially produced to meet these requirements.

The problematic of anomaly detection deals with being able to detect all anomalies without false alarms. This methodology, through extensive applications of NADA to documented traces of our database, make possible to evaluate the probability of detection and the probability of false alarms. They are depicted on Figure 5. These curves were obtained considering our documented traces with DDoS with different intensities, and varying the value of K in the range of]0, 3] when running NADA. A total of 12 different traffic traces were considered.

In plot P_D vs. P_F of Figure 5 the ideal set point (all attacks would be detected and no false alarm raised) is the left upper corner. The worst case is the diagonal, when

the results do not significantly differ from those obtained at random. Our curve clearly shows that NADA is efficient. Also in Figure 5, the plot P_D = f(K) and P_F = g(K) shows that whatever the value of K is, the detection probability is always higher than the rate of false alarms. In some way, this increases our confidence in the attribution of K = 2, as a starting value when using NADA.

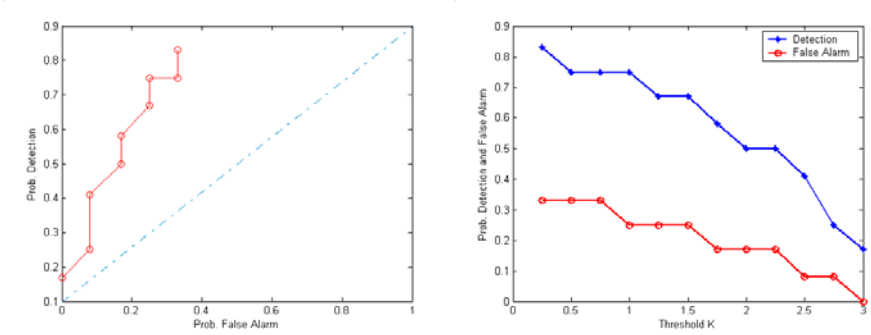


Fig. 5. Statistical performance. Left, Detection probability PD vs. False Alarm probability PF, PD = f(PF), right, PD = f(K) and PF = g(K).

6. Conclusion

In this paper, we have presented NADA an algorithm for detecting, classifying and identifying anomalies of any type in network traffic. NADA has been designed by putting together much work and results on anomaly detection previously published, but never integrated all together in a single algorithm. To complement other similar approaches, NADA provides information about the parties responsible of the anomaly, in a way easily understandable by technicians who are operating and managing networks.

In order to be efficient, and not restricted to just one type of anomalies, the strength of NADA deals with its multi-scaling and multi-criteria (in this paper we illustrate the algorithm with 3 criteria) analysis approach working on different (possibly all) IP aggregation levels. Despite this multi-dimensional analysis of traffic time series, the algorithm remains extremely simple. In particular, the power of NADA is related to its simple filtering parameter K, which allows even the detection of low intensity anomalies. Also important, is the way information provided by NADA is delivered: in graphical or textual formats, thus constituting an obvious anomaly database. If the first format could be interesting for administrator to discover, at a glance, what is happening in the network, the latter one could be easily used to trigger other types of signals or actions, suited to the anomaly that is occurring.

This paper also proposed an assessment methodology for anomaly detection tools. It is based on the evaluation of detection probability, vs. probability of false detections. NADA has been evaluated based on its capability to detect documented anomalies contained in real traffic traces. These anomalies have been artificially

generated in addition of the normal operational traffic of the Renater network. We tried as much as possible to create traces containing all kinds of anomalies. In addition, these anomalies have been generated with many different parameters, especially different intensity levels for assessing the capability of anomaly detection tools to detect both high and low intensity anomalies. For finalizing this work, we are in the process of evaluating other anomaly detection tools with this methodology, on our traffic trace database, as well as on unknown traces (i.e. traces for which we do not know about the presence of anomalies). We will then be able to compare their efficiency in detection as well as in classification. This work will be finished for the final version of the paper if it gets accepted. Future work also includes the design of a selection method for the K factor, as it is for the moment hand made.

References

1. Barford, P., Kline, J., Plonka, D., and Ron, A.: A Signal Analysis of Network Traffic Anomalies. In Internet Measurement Workshop, Marseille (2002)
2. Krishnamurthy, B., Sen, S., Zhang, Y., and Chen, Y.: Sketch-Based Change Detection: Methods, Evaluation, and Applications. In Proceedings of IMC'03, Miami (2003)
3. Hussain, A., Heidemann, J., and Papadopoulos, C.: A Framework for Classifying Denial of Service Attacks. In ACM SIGCOMM, Germany (2003)
4. Jung, J., and Krishnamurthy, B., and Rabinovich, M.: Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In WWW, Hawaii (2002)
5. Guo, X., Qian, D., Liu, M., Zhang, R., and Xu, B.: Detection and Protection Against Network Scanning: IEDP. In Proceedings of ICCNMC'01, Beijing (2001)
6. Kim, S., Reddy, A., and Vannucci, M.: Detecting Traffic Anomalies through Aggregate Analysis of Packet Header Data. In Networking 2004, Athens (2004)
7. Estan, C., Savage, S., and Varghese, G.: Automatically Inferring Patterns of Resource Consumption in Network Traffic. In ACM SIGCOMM'03, Germany (2003)
8. Lakhina, A., Crovella, M., and Diot, C.: Mining Anomalies Using Traffic Feature Distributions. In ACM SIGCOMM, Philadelphia (2005)
9. Claffy, K., Braun, H., and Polyzos, G.: A Parameterizable Methodology for Internet Traffic Flow Profiling. In Selected Areas in Communications – IEEE Journal, Vol. 13, (1985) 1481-1494
10. Cormode, G., and Muthukrishnan, S.: What's New: Finding Significant Differences in Network Data Streams. In IEEE/ACM Transactions on Networking, Vol.13, (2005) 1219-1232
11. MetroSec project. At http://www2.laas.fr/METROSEC/