An Anomaly Detection Model in a LAN using K-NN and High Performance Computing Techniques

Mercedes Barrionuevo*, Mariela Lopresti, Natalia Miranda, Fabiana Piccoli

LIDIC. Universidad Nacional de San Luis,
Ejército de los Andes 950 - 5700 - San Luis - Argentina

{mdbarrio, omlopres, ncmiran, mpiccoli}@unsl.edu.ar

Abstract. Detecting unusual values from large volumes of information produced by network traffic has acquired considerable interest in the network security area. Having a system of detecting anomalous events in a time near their occurrence, it is important for all computer systems in a network. Detecting anomalous values can lead network administrators to identify system failures, take preventative actions and avoid a massive spread. Anomaly detection is a starting point to prevent attacks. In this article, we present a form of data preprocessing to identify anomalies using a supervised classification algorithm, image processing, parallel computing techniques and Graphical Processing Units.

Keywords. Network Traffic. Anomaly. K-NN. High Performance Computing. GPU.

1 Introduction

Nowadays variety and complexity of internet traffic exceed everything imagined by Internet pioneers. We are currently immersed in a society dependent of computer systems, they are present in many areas such as: finances, industry, medicine and others aspects of everyday life. In order to protect or prevent these systems and their information (they can be important or relevant), it is necessary to implement technologies or models to avoid unauthorized or malicious access. This may be possible if valid access patterns are determined.

Threats to a data network are conformed by a set of frames with specific characteristics to look for vulnerabilities in a system. These vulnerabilities represent risks and can be used to perform future attacks.

When situations out of normal network profile are detected, administrators ask themselves questions such as What do they mean? Can such a situation be considered an attack? Does this deviation belong to traffic generated by new applications? Based on these, appear detection systems based on anomalies which report all unusual activity although they can be normal or not.

Generally, systems whose objective is to detect network attacks usually have a high error level, therefore it is necessary to have a semantic interpretation of their results.

This feature together to multiplicity of network traffic generated by different applications and characteristics such as: bandwidth, time of connections, among others, determine a work of high computational cost.

The most of researches in this area tend to evaluate the deviation of new instances respect to the normal profile of network traffic. Convert these results into semantics reports for network administrators is a big challenge. Based on this, we propose a Parallel-Supervised Network Anomalies Detection System, P-SNADS.

In [1], we present a first non-supervised P-SNADS model, it detects anomalies through images (they represent the network traffic) comparison by SIFT (Scale Invariant Feature Transform) [5]. In this paper, we propose to combine traffic classification techniques and high-performance computation to obtain good results when we work with large data volumes and the shortest possible time. For this, we focus on the preprocessing stage of data obtained from the network. A parallel supervised classification algorithm K-NN (K-Nearest Neighboring) is used to obtain a smaller data set to process in the second stage.

This document is organized as follows: the next section describes the theoretical concepts involved in development. Section 3 details the characteristics of the first stage of P-SNADS and section 4 shows the experimental results. Finally, the conclusions and future works are detailed.

2 Background

This work involves many concepts, among them we emphasize computers networks traffic and its anomalies, extraction methods of packages characteristics and supervised classification algorithms. In this section, we describe each one of them.

2.1 Computer Networks Data Traffic

Network traffic provides information about what travels by network. The most common data types are log data, such as Internet Protocol (TCP/IP) records, event logs, internet access data, Network Management Protocol (SNMP) data reporting, among others [10].

This information is necessary for network security, specifically for anomalous events detection. Fig. 1 illustrates an example of TCP/IP traffic, the rows detail individual network traffic and the columns are specific characteristics of each traffic. In the example, the first column is a session index for each connection, and the second says when the connection has occurred [10].

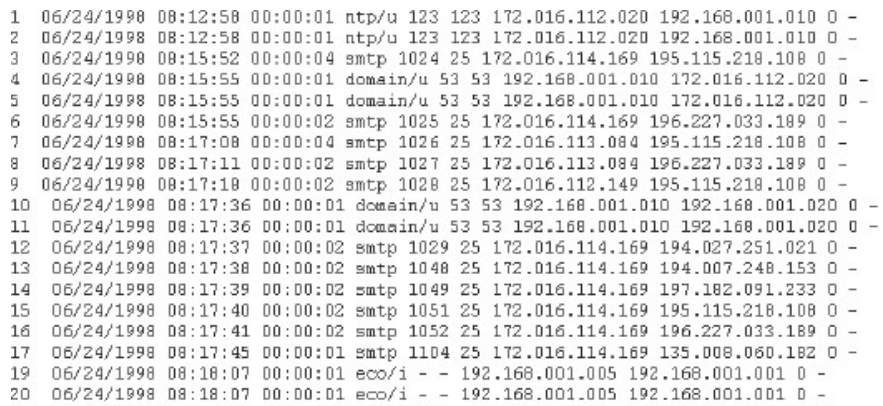

Fig. 1. Example of TCP/IP Traffic.

Data traveling on network can provide important information about user and system behaviors. These data can be collected with some commercial products or specific software, for example TCP/IP data can be captured using different tools, called sniffers.

Network traffic is composed of packets, flows and sessions. A packet is a data unit exchanged between a source and a destination on the Internet or another TCP/IP-based network; a network flow is a one-way packets sequence between two endpoints; and the session data represents communication between computers. A communication involves the interchange of multiple flows.

Traditionally, an IP flow contains a set of attributes, for this work, the more important are: IP address of source and destination, source and destination port, and protocol type. The protocol type, if you consider the 4-layer TCP/IP model, can be TCP or UDP (layer 3) or ICMP (layer 2).

This information allows to establish a behavior baseline or normal pattern of network traffic, and in consequence to identify unexpected or unwanted conduct, called anomalous traffic. Therefore, an analysis strategy by anomalies bases on the traffic description in normal conditions to classifies as anomaly all patterns that move away from it. In order to obtain this dataset, there are several techniques, some of which are mentioned in the following subsection.

Data Analysis of a Network Packet.

Studying the particular aspects of network traffic, it is necessary to extract only information from data packets and then process them. There are different techniques of extraction and processing, some of them are:

- *Graphical representation of raw data*: Generally, the representations are 2D and 3D scatter graphics, time-based graphics, histograms, pie charts, or diagrams.
- *Statistical information and pattern extraction:* They are based on average calculations, time distributions and probability distribution functions.

- *Analysis based in rule (signatures), anomaly detection and policy:* All traffic inspection analyzer that look for coincidences with a particular rule or signature belong this category. Rules are defined as values for certain fields in the header or a combination of several of them. These techniques are used in intrusion detection systems (IDS), such as Snort.
- *Flow-based analysis:* It focus on network traffic management as flow. The most of network information exchanged is oriented to connection (and non-oriented to packet), the analysis can take advantage of this. A clear example of typical network flow is a TCP connection, where the data exchanged are governed by the TCP state machine [8].

Each of these techniques is suitable for specific situations, also it is possible combine them. This work is based on flow analysis and rule-based analysis.

Usual Attacks.

One of the biggest challenges for network administrators is to detect attacks on computer networks. An attack implies to take advantage of a computer system vulnerability (operating system, application software, or user's system) for unknown purposes but, usually, causing damage. Therefore, it is impossible to make a complete classification of all the actual attacks and possible weaknesses of the networks, even more when networks are connected to Internet. The denial of service (DoS) and distributed denial of service (DDos) attacks are of great interest today.

A DoS attack comes from a single entity and its goal is to turn out unavailable the resources or services of a computer. There are different types, in particular this work has focused on the DoS attacks: Smurf, Fraggle and Land [3, 4], each one of them has the following characteristics:

- *Smurf:* This attack uses ICMP protocol (Control Management Protocol) to send a broadcast ping with a false source address. There are different ways to do a ping, they are:
    - *Normal Ping:* One or more ICMP echo requests are sent to a system, which responds with one or more ICMP echo replies. Thus, this operation verifies remote system.
    - *Broadcast Ping:* This ping sends an ICMP echo request to a broadcast address. Each system responds to sender, flooding it with ICMP echo replies.
    - *Broadcast ping with false source:* A broadcast ping is sent with victim's source address. Each system in network replies and floods the victim with answers. This operation is a combination of the two previous Ping.

  The pattern to recognize this attack type is to analyze to ICMP protocol, if source and destination IP addresses belong to the same network, and the destination address is a broadcast message.

- *Fraggle***:** When you want to check if a system is working, you can use UDP-based tools instead of ICMP to inspect whether the system is listening by a specific port

or not. This is commonly done with different types of vulnerability scans, that are used by attackers or security administrators. For example, if a system listens over Port 19 (TCP or UDP), when a connection is established, the system would respond with a constant character flow. Typically, the source system uses the TCP or UDP Port 7. When the source system begins to receive characters, it knows that the target system is operational and closes the connection. In a Fraggle attack, a broadcast packet is sent with a false address to the victim's port 19, if it has its port 19 open, it answers a constant flow of characters to victim. The pattern is similar to that of Smurf but in this case the protocol is UDP.

- *Land*: It's an attack using the TCP protocol. It creates an "infinite loop" which is caused by sending a SYN request with the same source and destination IP address. The victim computer responds to itself until a blocking state appears and it does not accept any new requests. Besides, as all processor resources are exhausted, the denial of service happens. To recognize this type of attack, it is necessary to analyze coincidence between the source and destination IP addresses, as well as the same ports.

As mentioned above, there are many other denial of service attacks, but the detection of patterns in them requires a deeper and detailed analysis that escapes this work.

2.2 Supervised Classification Algorithm K-NN

The classification process builds models capable of determining if an object, from its characteristics, is member or not of a category. The classification is supervised when, in advance, there is an already classified observations set, and it is known to which set belongs each observation. The algorithms dedicated to solve the supervised classification problems usually operate with information provided by a set of samples, patterns, examples or training prototypes that are assumed as representatives of each classes [9].

In particular, this paper uses a supervised classification algorithm based on neighborhood criteria. It is known as K nearest neighbor (K-NN). The method of the nearest neighbor and its variants are based on the intuitive idea: "Similar objects belong to the same class". The class can be determined by comparing the object against all elements in a class. The selected class is that holds the most similar objects.

The similarity idea is formally reflected in distance concept, usually the Euclidean distance is used [6]. The calculation of the nearest neighbor can be solved using parallel techniques, in [6] has been shown that the parallel implementation using GPU [7] obtains very good response times.

3 Parallel-Supervised Network Anomalies Detection System

The detecting task of possible packets with anomalous data in a computer network is very expensive. A good alternative can be combining classification techniques and high-performance computation (HPC) in a only one solution. P-SNADS is a system that combines HPC, supervised classification and image processing to detect possible attacks. In P-SNADS' architecture, we can distinguish two stages, the first one makes all necessary steps to capture and pre-classify traffic, while the second one is respon-

sible to detect possible attacks. In this stage, the decision is based on images comparison all of them generated from data obtained in the previous stage. In this work, we focus on the first step of system, the second was presented in [1].

The stage 1 captures network traffic, pre-process and organizes it in data flows. From these flows, P-SNADS determines whether each one of them is normal or possibly anomalous. Fig. 2 shows a P-SNADS graphical representation.

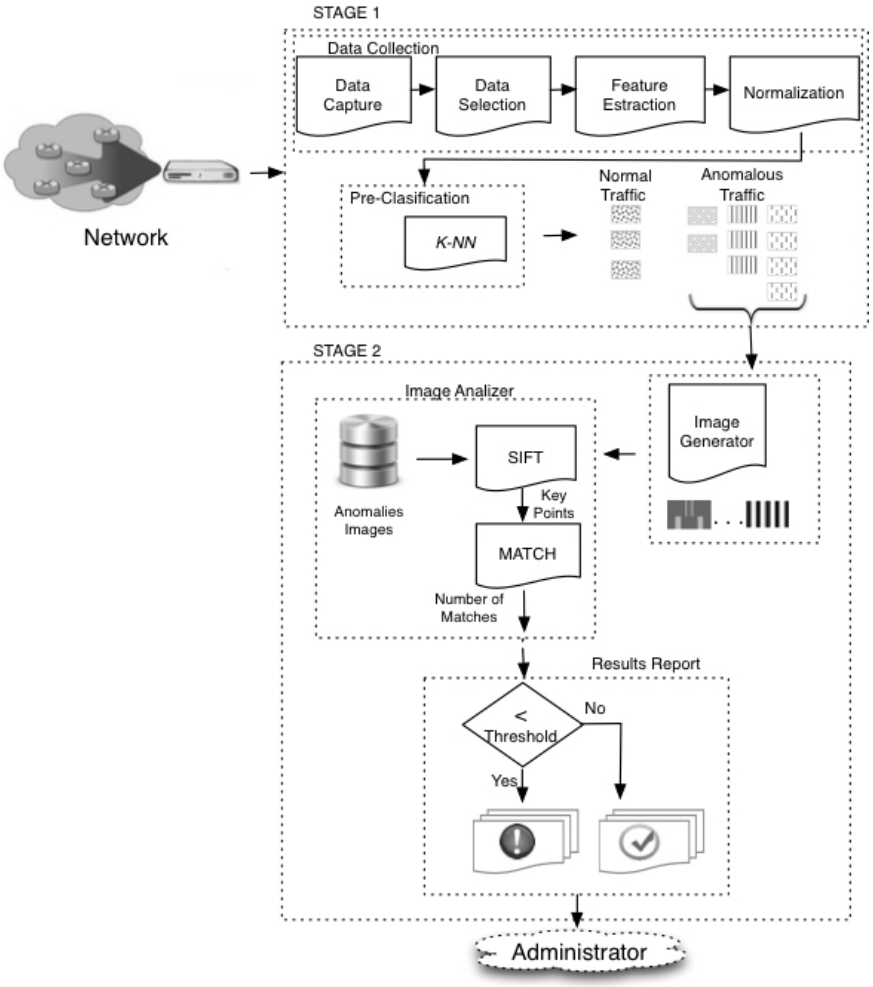

Fig. 2. P-SNADS' Architecture.

The stage 1 groups different tasks, we distinguish two tasks: data collection and data pre-classification, each has the following features and functions:

- *Data Collection:* Its objective is to obtain data to work, classify and reduce its volume for the Data Pre-classification. Data Collection implies the following subtasks:
  - *Data Capture:* This process catches network traffic. It uses a sniffer to do it, and it is activated at specific moments, for examples when there is more network traffic.
  - *Data Selection*: This step selects frames to be analyzed, according to the attacks considered, these are TCP, UDP and ICMP frames.
  - *Feature Extraction:* In this step, we extract of each frame all interest fields to be analyzed. They are: source and destination IP address, source and destination port and protocol type (TCP, UDP or ICMP).

  Fig. 3 shows examples of tuples of normal traffic patterns (a) and anomalous traffic (b).

$$f_{normal1} = \{\ 10.0.0.1,\ 212.48.72.19,\ 31215,\ 80,\ UDP\}$$
$$f_{normal2} = \{\ 10.0.0.1,\ 13.29.10.199,\ 2233,\ 25,\ TCP\}$$
$$f_{normal3} = \{13.29.10.199,\ 10.0.0.1,\ ,\ ,\ ICMP\}$$

(a) Normal traffic patterns.

$$f_{Srmurf} = \{\ 10.0.0.1,\ 10.255.255.255,\ 31245,\ 80,\ \text{ICMP}\}$$
$$f_{Land} = \{\ 10.0.0.1,\ 10.0.0.1,\ 80,\ 80,\ TCP\}$$
$$f_{Fraggle} = \{\ 10.0.0.1,\ 10.255.255.255,\ 31245,\ 80,\ \text{UDP}\}$$

(b) Anomalous traffic patterns.

Fig. 3. Normal and anomalous traffic examples.

  - *Normalization:* This step is fundamental, the tuples become an integer values vector. All address IP (IPv4) a.b.c.d are transformed according to equation (1).

$$(a \times 256^3) + (b \times 256^2) + (c \times 256^1) + (d \times 256^0) \quad (1)$$

- *Data Pre-classification:* Threats are not the only traffic on network, all packets generated in an attack coexist with regular traffic packets. This module classifies the frames obtained at Data Collection step as normal or anomalous traffic. To make this, we used the K-NN classification algorithm, it is implemented in parallel using GPU. Its work is to compare each vector (pre-processed flow), with every data flows recorded and representative of an anomaly. In consequence, a distance is computed to determine whether or not each new data flow can be considered an anomaly.

The following section analyses the performance of the P-SNADS, considering the effectiveness in anomalies detection.

4 Experimental Results Analysis

This section presents a experimental results analysis for the first stage of P-SNADS. T-Shark tool is used to catch data traffic. We work with several samples, each of them has approximately 2000 frames. These frames belong to a local area network (LAN) of Networks Laboratory of Universidad Nacional de San Luis. The three attacks are simulated in a server and they pretend to deny the HTTP service.

Once obtained the standard frames, we build different databases, all of them have normal and anomalous traffic. The Pre-classification module applies K-NN, it receives the databases as input data and performs an evaluation for different values of K. We consider four values of K, they are 5, 7, 10 and 12.

To compute K-NN, we use a computer with a GPU Tesla K20c (processors quantity= 2496, memory size= 4.6 GB, clock frequency= 706 MHz and processor memory clock= 2600 MHz).

When K-NN are computed for every different value, we evaluate the experimental results considering the following metrics: Positive Predictive Value (Precision - PPV), True Positive Rate (Recall - TPR), and F-measure (F) [2]. All of these measures can be calculated based in the next parameters:

- *True positive (TP):* It is the number of anomalous frames correctly detected.
- *False positive (FP):* It is the number of normal frames wrongly detected as anomalous frames.
- *False negative (FN)*: It is the number of anomalous frames not detected.
- *True negative (TN):* It is the number of normal frames correctly detected.

From these parameters, the metrics are defined as:

- *Positive Predictive Value or Precision:* it represents the possibility that a sample labeled as positive is indeed a true positive. It is the percentage of detected frames that are actually anomalies. Mathematically defined as:

$$PPV = \frac{TP}{(TP + FP)}$$

- *True Positive Rate or Recall:* It is the percentage of actual anomalous frames that are detected. It is calculated according to:

$$TPR = \frac{TP}{(TP + FN)}$$

- *F-measure (F):* It is a typical information retrieval metric, defined as the harmonic mean between the two-metrics explained above: Precision and Recall. Generally, F is used as a performance metric of K-NN in anomalies detection. Its formula is:

$$F = 2 \times \frac{(PPV \times TPR)}{(PPV + TPR)}$$

The results obtained with each metric are shown in Fig. 4. In it, each attack: Smurf (a), Land (b) and Fraggle (c) are considered.

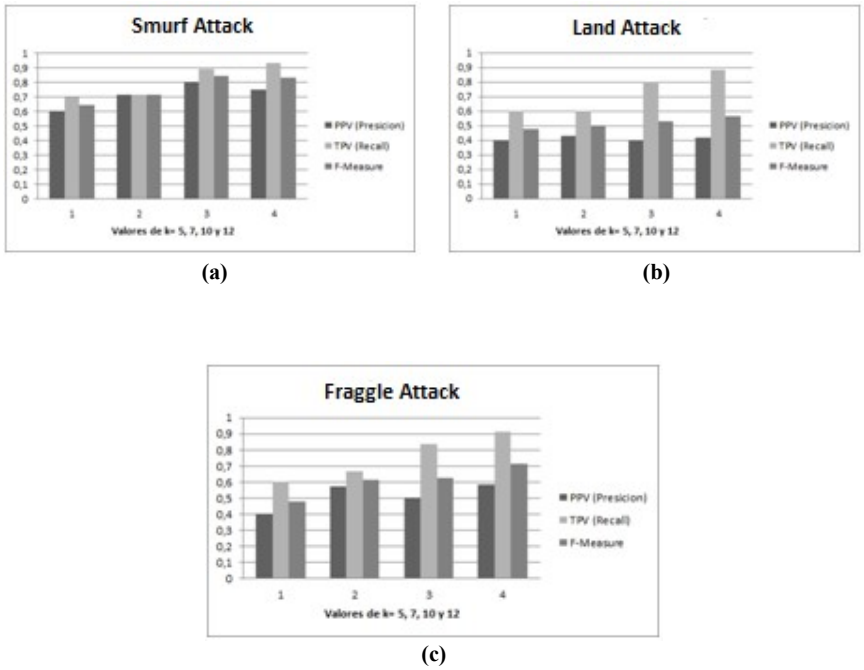

Fig. 4. Results obtained in attacks: *Smurf, Land, Fraggle*.

From observation of the above graphs, we can say that Smurf attack has an approximate accuracy of 72%, while for the others two are lower: Land= 41% and Fraggle = 52%.

Regarding Recall, more significant results are obtained, to Smurf is 80%, Land is 72% and Fraggle is 75%. In base of these values, we can infer that the detection rate of anomalous traffic is high, particularly when K = 12.

Finally, the F-measure also returns good results, particularly for K = 12 no matter which attack is. The values obtained are 0.83 for Smurf; 0.56 for Land and 0.71 for Fraggle. It means that our model performs well in detecting attacks.

Therefore, if we consider the metrics analyzed, we can conclude that P-SNADS satisfactorily achieves the objectives of this work.

5 Conclusions and Futures Works

Today, the anomalous traffic detection is a task of great interest, although there are several detection tools, there is a lot of work to be done. This obeys to large amount

of circulating data on the Internet and constant change of traffic profile.

This work is focused on the traffic classification stage; we propose a model and analysis its feasibility for three known attacks, all of them are of kind of services denial: Smurf, Land and Fraggle. In order to accelerate the process and obtain results in less time, the implementation is solved using HPC techniques in GPU.

The proposal is evaluated according to different metrics, the proposed model shows an accuracy between 40% and 70%, and sensitivity between 60% and 83%. In addition, F-measure is used to measure the system performance, obtaining values between 0.5 and 0.83.

Although the results are satisfactory, as a future work it is necessary to analyze which factors affect and produce different performances according to attack. Another future job is to evaluate P-SNADS for others attack patterns. It is also intended to compare our development with those that use machine learning techniques or intelligent tools.

References

1. Barrionuevo, M., Lopresti, M., Miranda, N., Piccoli, M.: Un enfoque para la detección de anomalías en el tráfico de red usando imágenes y técnicas de computación de alto desempeño. XXII Congreso Argentino de Ciencias de la Computación. CACIC 2016. p. 1166-1175 (2016)
2. Davis J., Goadrich, M.: The relationship between precision-recall and roc curves, in ICML '06: Proceedings of the 23rd international conference on Machine learning. New York, NY, USA: ACM, 2006, pp. 233–240 (2006)
3. Gibson, D.: CompTIA Security+: Get Certified Get Ahead: SY0-201 Study Guide Createspace Independent Pub (2009). ISBN 9781439236369.
4. Henao Ríos J. L.: Definición De Un Modelo De Seguridad En Redes De Cómputo, Mediante El Uso De Técnicas De Inteligencia Artificial. Tesis presentada como requisito parcial para optar al título de Magíster en Ingeniería – Automatización Industrial. Universidad Nacional de Colombia. (2012)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. International journal of computer vision. Pp 91-110, (2004)
6. Miranda, N.: Cálculo en Tiempo Real de Identificadores Robustos para Objetos Multimedia Mediante una Arquitectura Paralela GPU-CPU. Tesis de Doctorado en Ciencias de la Computación. UNSL (2014).
7. Piccoli María F.: Computación de alto desempeño de GPU. 1era edic. ISBN: 9789503407592. La Plata Edulp, (2011)
8. S. Institute, Transmission Control Protocol: DARPA Internet Program Protocol Specification. Defense Advanced Research Projects Agency, Information Processing Techniques Office, (1981).
9. Tribak Hind. Análisis Estadístico de Distintas Técnicas de Inteligencia Artificial en Detección de Intrusos. Tesis Doctoral.Universidad de Granada. (2012).
10. Wang Y.: Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection, Chapter III Network Traffic and Data, Information Science Reference - Imprint of: IGI Publishing, (2008).