Ciência e Natura
ISSN: 0100-8307
cienciaenaturarevista@gmail.com
Universidade Federal de Santa Maria
Brasil

Bateni, Shirin; Asghar Khavasi, Ali

Design a security firewall policy to filter incoming traffic in packet switched networks using classification methods

Ciência e Natura, vol. 38, núm. 2, mayo-agosto, 2016, pp. 821-830
Universidade Federal de Santa Maria
Santa Maria, Brasil

Available in: http://www.redalyc.org/articulo.oa?id=467546204023

Artigo Original *DOI:10.5902/2179-460X21530*

Ciência e Natura, Santa Maria v.38 n.2, 2016, Mai.- Ago. p. 821 –830
Revista do Centro de Ciências Naturais e Exatas - UFSM
ISSN impressa: 0100-8307 ISSN on-line: 2179-460X

CIÊNCIA e NATURA

Design a security firewall policy to filter incoming traffic in packet switched networks using classification methods

Shirin Bateni and Ali Asghar Khavasi

Faculty Member, Computer and IT Engineering Department, Islamic Azad University, Zanjan Branch, Iran

Abstract

Firewalls are core elements in network security. However, managing firewall rules, especially for enterprise networks, has become complex and error-prone. Firewall filtering rules have to be carefully written and organized in order to correctly implement the security policy. In addition, inserting or modifying a filtering rule requires to overcome and filter a range of special attacks or issues in network. In this paper, we present a machine learning based algorithm that filter Denial of Service (DoS) attacks in networks. This filtering algorithm has been designed by using a classification algorithm based on principal component and correlation based filters. We show good quality and performance of our algorithm experimentally by executing our algorithm on a several packet flow data sets.

Keywords: *Firewall. Denial of Service Attacks. Machine Learning. Classification.*

Recebido: 21/03/2015 Aceito: 22/04/2016

1 Introduction

Connecting to the Internet without a firewall is a bit like parking your car to go to shop while leaving the doors unlocked, windows rolled down and keys in the ignition (Al-Shaer, 2014). While you may be able immediately do a good response in the event of theft, but have created a valuable opportunity for the thieves to achieve their destructive goals as quickly as possible. Such situation exists on the Internet and attackers at the first stage identify their victims by using malicious code such as viruses, worms and Trojans and then attack the identified targets at later stage. A firewall application offers an appropriate level of security and protection against these types of attacks (Van Raamsdonk, 2014).

In fact, a firewall is software that acts like a protective wall between your computer and the Internet. It allows you to surf the internet easily and feel secure during your activities. There are two types of firewall, some are software which running on a computer to protect it and others are hardware which protecting the entire network (Sheth et al., 2014).

Nowadays by enhancing and extending different firewalls; different features and technologies are proposed to protect computer networks against attacks and damages. Security policies of a firewall are the core and the most important parts in a firewall; policies such as virus protection, malware protection, cyber-attack protection and etc. An important section in a security policy, especially in network firewalls, is policies to prevent network cyber-attacks. This type of attack has a wide range of attacks; including Denial of Service, Distributed Denial of Service, etc (Antikainen, Aura, & Särelä, 2014).

To deal with this kind of cyber-attacks which is imposed through network communications, identifying the stream flows through network or firewall is important. Therefore, in this paper we will concentrate on detecting and identifying inbound flows using learning machine. To do this, we go on elaborating the concept of the network flow and explaining some of these attacks (Jun, Kim, Cho, Ahn, & Kim, 2014).

In packet-based networks, traffic flow, packet flow or a network flow is a sequence of packets from a source to a destination computer, is sending information to several networks simultaneously. RFC2722 protocol defines traffic flow as the equivalent of a call or logical connection (N Brownlee, C Mills, & G Ruth, 1999). RFC3697 protocol defines it as sending a sequence of data packets from a specific source to a specific destination or group of computers (Kazantzidis, Gerla, & Lee, 2001). Flow may include all packets of a transferred data or media stream. Also RFC3917 protocol defines flow a set of data packets crossing a network in period of time (Rajahalme, Amante, Jiang, & Carpenter, 2011).

1.1 Denial of Service Attacks

Denial of Service (DoS) Attacks, in network security concept, defending and protecting against the attacks is too important. Denial of Service or DoS attack is an attempt to make a machine or network resource unavailable to its legal users, such as to temporarily or indefinitely interrupt or suspend service of a host connected to the Internet (Eckhardt, Mühlbauer, AlTurki, Meseguer, & Wirsing, 2012). Although different motivation may be the cause of this type of attacks, attempting to make a service temporary or permanently unreachable is its common goal. Criminal perpetrators of DoS attacks often target sites or services hosted on high-profile web servers such as banks, credit card payment gateways; but motives of revenge, blackmail or activism can be behind other attacks. One of the common attacks is saturating target machine with external communications requests, such that it cannot respond to legitimate traffic or response to be slow or unavailable. Such attacks lead to extra burden on server. DoS attack enforce the aim computer to reset or use its resources, therefore it would be unable to service to its already considered services and also this attack transgresses policies that are acceptable for Internet service providers. Emblems of denial of service define in this ways (Mirkovic & Reiher, 2004):

Slow and abnormally function of network- being unavailable for a specific website- excessive accretion of numbers in received spams- disconnection of Internet. Denial of Service could lead to some problems in attacked computer which lead to risk the referred computer or the whole network and other computers in LAN. If attack had happened in

considerable scale, all the geographical areas of internet connections, due to incorrect configuration or loose infrastructures of equipment, network would face a challenge (Pelechrinis, Iliofotou, & Krishnamurthy, 2011).

1.2 ICMP Attacks

Internet Control Message Protocol (ICMP) flood, smurf attack, is one of those flood-like DoS attacks on internet. This kind of attack depends on unsuitable configuration of network equipment which let sending the pockets to all host computers on a specific network which give all contribution addresses. In such attack, invaders with a fabricated IP, send a request to check (ping) one or several all- contribution servers, then they loosen the aim machine (sacrifice) IP address. All-contribution server sends this request throughout network. All the network machines send the reply to server in the form of all-contributed sending. All-contributed server sends or leads received replies towards aim machine. Thus whenever invader machine all-contributes a request to several servers on different networks, the collection of all the computer replies of different networks would be sent to aim machine and incapacitates it. So network broadband uses rapidly and the transformation of permitted pockets impedes. To fight against denial of service in internet, services like Smurf Amplifier Register provides the ability to recognize unsuitable configuration of network and also capability of suitable process like filtering. Flood-like Ping pockets, send a lot of Ping pockets towards sacrifice. It is pertinent to sending of Ping pockets in an adverse form or position towards sacrifice which leads operating system to crash (Hadi, Azmat, & Ali, 2013).

1.3 SYN Attacks

SYN Flood, SYN Flood happens when a host uses one of SYN Flood pockets (SYN/TCP), and the sender address is wrong. Whichever of these pockets, are like a request for connection and cause server to be involved in several half-opened connections, but due to the wrong address no replies would returned. These half-opened connections, saturate the number of connections in server availability and make it unable to respond to permitted requests until the end of attack. Therefore server resources would be allocated to half-opened connections and would be disabled to respond to server requests (Bogdanoski, Suminoski, & Risteski, 2013).

1.4 Teardrop Attacks

Teardrop attack, this attack includes sending of those IP packets which are confused or IP packets in larger size or disorganized packets. This attack could cause different operating systems to crash, due to the disturbance in revival of recoding in TCP/IP parts. Windows 3.1x and Windows 95 and operating system of NT Windows and 2.0.32 & 2.1.63 copies are vulnerable in exposure of this attack (Darwish, Ouda, & Capretz, 2013).

1.5 Peer-to-peer Attacks

Peer-to-peer attacks, Attackers are searching for finding dysfunctions in Peer-to-peer servers to initiate DoS attack. Peer-to-peer attacks are distinguished from those attacks based on bot-nets. In Peer-to-peer attack, there is not a bot-net or client to communicate, instead there is an invader which places as a representative of master who teaches existent clients in file sharing centers to disconnect their peer-to-peer network and instead connect to sacrifice website. Consequently, myriads of computers would be connected to sacrifice website. While web server is able, before decreasing of its functionality, to control and manage hundreds of connections per second, it fails immediately after 5 or 6 thousands connections per second (Fiandrotti, Gaeta, & Grangetto, 2015).

2 Related Work

The aim of this part is, first of all to define and exact analysis of words and extant terminologies in this paper in order to further illumination of the issue, secondly analysing and representing of previous projects about identification of the kind of input flow.

Classification, this is a process, in which objects, regarding their already defined characteristics would be divided and each piece ascribes to a special class, known as classification.

In this process, firstly the main and important characteristics of objects would be recognized and according to that, some categories of objects would be defined then in the test phase another new object would add to category and this

category based on its features, acts to ascribe the new object into an already defined category or class.

The concept of classification is one of those critical concepts in learning machine. In this topic there are valid and famous algorithms that among them we can refer to algorithm C4.5, Decision Tree, Nearest Neighbour, Bayesian Network, and Support Vector Machine, which have been applied and used in various conditions and processes.

Classifications of Network Traffic, in this process, according to the main and important characteristics existing in Network Traffic, first some categories of traffic like TCP, UDP, FTP, Virus, Attack and ....would define then in test phase, every input flow and traffic, based on amounts that they contain for different features, would be ascribed to one of the classes, thus the kind of traffic reveal.

After explaining of above concepts, we are going to represent previous projects related to this topic.

In paper (Callado et al., 2009) a superficial review on detective and dividing methods of input traffic of computer networks have taken place. Methods have been divided into the type based on packet and the type based on flow and each one of these methods is explained briefly and the results of them and their comparisons to each other are represented. This paper has pointed many methods based on learning machine that have been considered in most of these classifying techniques papers.

In paper (Cireşan, Meier, Masci, & Schmidhuber, 2012) in order to classifying of network flows, nervous network has been used. The action that has been done in this paper is designing a multi-layered and multicolumn nervous network to recognize the kind of input flow. After designing, the network gives a considerable amount of instructing data to nervous network in order to learn and when finally the nervous network obtains enough quality of accuracy, it would be used as a testing device for network flow of data. This technique is very significant due to its high speed, but there is a problem in this method and that is the lack of dynamism. That is, in order to add another qualification to network, all the time consuming process of nervous network must be repeated.

In paper (Zaklouta & Stanciulescu, 2014), classification technique has been used in order to determine the kind of input traffic classification. The aim and the final idea of this paper is to improve the quality of service. In other word, in this paper the writer has tried, by using of classification technique based on the quality of service represented by network, to determine the kind of input traffic immediately in communication networks. Finally, acquired results of this method by using of ordinary line and schedule time technique in a common memory, have been analysed and compared.

In paper (Yu & Liu, 2003), the classification technique has been used in order to enhance the availability of network during distributed denial of service. The main attitude of this paper is applying of formal and main algorithms to identify a flow as leading flow into denial of service situation. The results of this method show, despite the influence of denial of service, the network identifies and represses the attack assuredly and the availability remains untouched. At the end of project, this method has been compared with other methods in respect of availability and also results show the better functionality of this method in this scope. In continue of this paper in part 3, we are going to explain the method and represented algorithm. In part 4, we will first go to the environment and testified collection of data and then representing of the implementing results of algorithm.

3 Proposed Method

In this part we aim to design a safety policy in firewalls to classify and filter the input traffic. The pertinent policy of filtering will be on flow surface not on packet surface. Therefore, designed mechanism must identify whole flow, as good or bad flow. If the flow is good then it could pass, if not so it won't be permitted to pass. Overall structure of this policy has brought below.

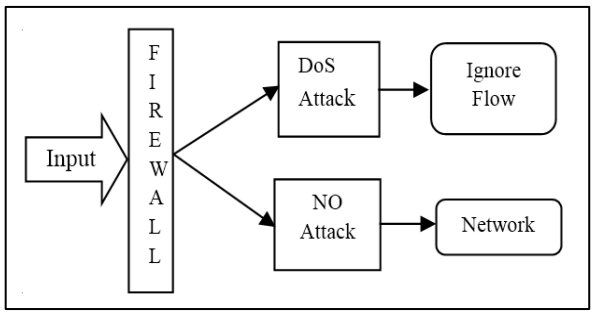

Figure 1 - overall structure policy of filtering the input traffic

The main idea and attitude that will be regarded in this paper is how classification flows will be identified. As mentioned in above figure, the kind of process will be applying the design policy inside the firewall and it will evaluate every input flow in respect of being safe and not being safe. Thus if safe it could flow into firewall and if unsafe the permission won't happen.

To designing of this filtering policy inside the firewall, following points are very important and should be observed in designing process:

(1) Since firewall should be fast and not being a burden on system, this filtering policy should be very fast.

(2) Since input flowing might contain fruitful and sometimes necessary data for the reciever, therefore this filtering policy must act absolutely accurate. In other word, it should not recognize a safe flow as unsafe, or the other way round.

Considering two points mentioned above, a very fast and absolutely accurate filtering policy is completely necessary and vital. The main concern and idea of this paper is identifying and filtering the input traffic by applying classification methods in order to distinguish safe and unsafe flows. In this paper, an approach to categorising and classification of input flows based on their main characteristics is proposed which then will determine, depending on to which category each flow belongs, as being safe and unsafe.

The main goal of categorizing and classification is to identify the type of input flow. Regarding whatever presented in part (Timofte, Zimmermann, & Van Gool, 2014) and (Nevil Brownlee, Cyndi Mills, & Greg Ruth, 1999), in order to classification of input flows, different features would be considered. The most significant features have been pointed above are according to Table 1.

Altogether input traffic or networks flow are divided as Table 2.

Finally, the represented algorithm should ascribe each input flows of new networks to one of above classifications.

In using learning machine for categorising of input traffic, there are two main and significant points:

1. How to apply the sets of practical features
2. Using which learning machine algorithm to classification of input traffic.

To simplifying the main issue, the classification of the network traffic, would be presented as a formal and mathematical form, showing below;

Table 1 - list of features for network flow

ID	The flow discriminator
1	Total number of packets in the flow
2	The average packet size of flow
3	The number of packets sent for the flow
4	The average send packets size of a flow
5	The variance of send packet's size
6	The average receive packets size of a flow
7	The variance of receive packet's size
8	The variance of received packet's size
9	The duration of the flow
10	The protocol (TCP or UDP)
11	The source port of a flow
12	The destination port of a flow
13	The number ratio of a send and receive packets
14	The byte ratio of send and receive packets
15	The number of SYN packets
16	The number of RST packets (rst)
17	The number of FIN packets (fin)
18	The average window size (window size)
19	The variance of window size

Table 2 - the kind of input flow into network

Number	Traffic class	Representative Applications
1	Bulk	ftp
2	Interactive	ssh, telnet, rlogin
3	Mail	pop3, smtp, imap
4	Service	X11, dns
5	WWW	http, https
6	P2P	Kazza, BitTorrent, Gnutella
7	Attack	worm, virus
8	Others	Scan, netbios, ntp, tsp

Suppose the set of x $T = \{t_1, t_2, ..., t_k\}$ would be the collection of network traffic classes and

the collection of $X = \{x_1, x_2, ..., x_k\}$ the collection of input flows with unidentified class. The main aim is finding a function like F.X=T which each member of collection X consider as a member of collection T.

In represented method to lessen the number of features, we use Fast-Correlation Based Filter algorithm that was presented in paper (Yu & Liu, 2003). Then by applying the analysis of the main factors on this reduced collection of features, we act to classifying of input traffic.

First of all we explain the Fast-Correlation Based Filter method. The semi-code of this algorithm based on paper (Yu & Liu, 2003) is like Figure 2.

As it is obvious in the figure too, this input algorithm is the collection of train data and the output algorithm is the collection of well-functioned features.

Consider that the amount of δ is a stable and also already defined.

Before describing algorithm, first of all we represent the existent parameters in it:
Unstable entropy X: (X is a vector)

$$H(X) = -\sum_j P(x_i)\, log_2(P(x_i)) \quad (1)$$

That $P(x_i/y_i)$ is a likely consequence for the amounts of x_i, if Yi is seen.

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_i) log_2(P(x_i|y_i)) \quad (2)$$

The amount of information acquiring which representing the reduction of X data in the case of observing Y, would be defined as below:

$$IG(X|Y) = H(X) - H(X|Y) \quad (3)$$

And finally the symmetrical uncertainty for both of X and Y variables would be defined as below:

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \quad (4)$$

So the amount of $SU_{i,c}$ would be measured as an amount of SU which is the amount of correlation between F_i and class C.

```
input:  S(F1, F2, ..., FN, C) // a training data set
        δ                     // a predefined threshold
output: Sbest                 // an optimal subset
1   begin
2     for i = 1 to N do begin
3       calculate SUi,c for Fi;
4       if (SUi,c ≥ δ)
5         append Fi to S'list;
6     end;
7     order S'list in descending SUi,c value;
8     Fp = getFirstElement(S'list);
9     do begin
10      Fq = getNextElement(S'list, Fp);
11      if (Fq <> NULL)
12        do begin
13          F'q = Fq;
14          if (SUp,q ≥ SUq,c)
15            remove Fq from S'list;
16            Fq = getNextElement(S'list, F'q);
17          else Fq = getNextElement(S'list, Fq);
18        end until (Fq == NULL);
19      Fp = getNextElement(S'list, Fp);
20    end until (Fp == NULL);
21  Sbest = S'list;
22  end;
```

Figure 3 - pseudo-code Fast Correlation Filter algorithm

Lines of 2-6:

In these lines the amount of SU would be measured for all F_i features. Each amount that would be more than the amount of threshold (mathematical term) would be add to S_{list}.

Line7:

S_{list} organizes in an ascending form.

Line 8:

Put the first element of the S_{list} into variable.

Lines 9-20:

Considering those Heuristics which have been defined for algorithm, in these lines all the organized elements of S_{list} would be analysed in order to identify redundant features and essential features, and at last to eliminate the redundant features from the S_{list}.

Line 21:

Put the edited list of the previous stage into the input variable(S_{best}).

Up to this time we reduced the networks' traffic of features list by applying above algorithm by using the best and the most practical way. In other word, we eliminate the redundant features that are not helpful in the process of classification in order to reduce the calculation and procession time in the classification stage.

Now we use following formula for classification by using the analysis of main factors method:

First we calculate the amount of the specific vectors of train data by using main factors in following steps.

Step 1)

We suppose that the train collection includes $X_1,X_2,.....X_n$ and each data includes π, that is, the learning matrix is a $X_{n\times p}$ matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} = [X_1 \quad X_2 \quad \cdots \quad X_p] \qquad (5)$$

The mathematical formula

Step 2)

We calculate data average:

$$\mu = \frac{1}{p}\sum_{i=1}^{p} X_i \qquad (6)$$

Step 3)

Calculation of data variance

$$\varphi_i = X_i - \mu \qquad ,1 \le i \le n \qquad (7)$$

Step 4)

Calculation of covariance from train data:

$$C = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)(X_i - \mu)^T = \frac{1}{n}\sum_{i=1}^{n}\varphi_i\varphi_i^T = \frac{1}{n}AA^T \quad ,A = [\varphi_1, \varphi_2, \dots, \varphi_n] \qquad (8)$$

Step 5)

Suppose the pairs of (mathematical formula) are vector pairs and specific amount of covariance matrix C.

In this stage we select the specific vector k which contains the most special amounts and put it in matrix U;

$$U = (u_1, u_2, \dots, u_k) \qquad (9)$$

So matrix U is a matrix of $U_{p\times k}$

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_p) \qquad (10)$$

Considering above descriptions, calculating of semi-code algorithm of the main factors is something like Figure 3.

Algorithm Essential_Component_Decomposition(ECD)

Input: train matrix $X_{n\times p}$

Output: eigen-vectors and eigen-values of covariance matrix λ, u

1. Calculate the mean of matrix X
$$\mu = \frac{1}{p}\sum_{i=1}^{p} X_i$$
2. Calculate the derivation from the mean:
$$\varphi_i = X_i - \mu \qquad ,1 \le i \le n$$
3. Calculate the covariance matrix of matrix X:
$$C = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)(X_i - \mu)^T = \frac{1}{n}\sum_{i=1}^{n}\varphi_i\varphi_i^T = \frac{1}{n}AA^T \quad ,A = [\varphi_1, \varphi_2, \dots, \varphi_n]$$
4. Make a matrix λ, u from k greatest eigen-values and eigen-vectors of C

Figure 3 - calculating of semi-code algorithm of the main factors

Now regarding above descriptions, the main algorithm to identify a kind of class of networks flow is like Figure 4.

Flow Classification Algorithm

1. Use fast-correlation based filter algorithm to remove redundant feature from feature set.
2. Calculate the matrix λ, u of train data using Essential Component Decomposition (ECD) algorithm.
   So we reduced the dimension of train data from p to k where $(p \ll k)$.
3. For each testing vector(flow) y we project it into k-dimensional subspace:
$$\hat{y} = uu^T y$$
4. For a projected testing vector(flow) $\hat{y}$ find the minimum ε_i to all of flow class I means:
$$\varepsilon_i = \|\hat{y} - F_i\|$$
5. For each i that ε_i is minimum, the flow is identified as a class i.

Figure 4 - pseudo -code algorithm to identify a kind of class in network flow

4 Experiments

The chief way of evaluating and testing, represented by Waikato University's library would be done (Alcock, Lorier, & Nelson, 2012). This library that with programming language C, has been put under UNIX implementation, would provide dramaturgy of network flow and processing on online packets. The applied data, also come from this university library. In following tables, necessary tools and the collection of essential data for testing and represented algorithm evaluating have represented.

In continue, the result of algorithm implementation on one of the data collection example would be represented and at last the total results of algorithm implementation on the

In Table 3 the overall characteristics of environment and design test tools have represented.

Table 3 - overall characteristics of environment implementation algorithm

Operating System	Ubuntu 15.04
Programming Language	Unix C
Libraries	Libtrace
Packet Format Supported	PCAP trace file PCAP interface ERF trace file DAG live capture

In Table 4, the tools and data sets are used by Waikato University's library to test the design have represented.

Table 4 - library tools list

Tool Name	Description
traceanon	anonymises trace files
traceconvert	converts a trace from one format to another
tracediff	reports differences between two trace files
traceends	summarises traffic sent and received by endpoints
tracefilter	applies a BPF filter to a trace
tracemerge	merges multiple trace inputs into a single trace
tracepktdump	displays packet contents in a readable format, similar to tcpdump
tracereplay	replays a trace file using original timing

Table 4 - continuation...

Tool Name	Description
tracereport	produces a variety of reports on a trace
tracertstats	produces stats about an input trace in real time
tracesplit	splits trace files
tracesplit_dir	splits trace files based on packet direction
tracestats	summarises number of bytes and packets matching BPF filters
tracesummary	summarises the basic stats for a trace
tracetop	reports the busiest flows over time, similar to ntop
tracetopends	reports the busiest endpoints in a trace

In Table 5, the data sets that are used by Waikato University's library to test the design have represented.

Table 5 - data sets list

Name	Duration (minute)	Packets (million)	Size (MB)
20100106-030946-0.dsl	20	23	743
20100106-033000-0.dsl	30	34	1098
20100106-040000-0.dsl	30	36	1,131
20100106-043000-0.dsl	30	35	1,104
20100106-050000-0.dsl	30	34	1, 094
20100106-053000-0.dsl	30	35	1,129

overall collections of above data would be compared to each other.

In this part the amount of algorithm accuracy in each collection of data has been represented in a percentage form to make it more understandable. The algorithm accuracy, in fact, is representing the degree of how successful has been the introduced method in identifying the kind of input flow.

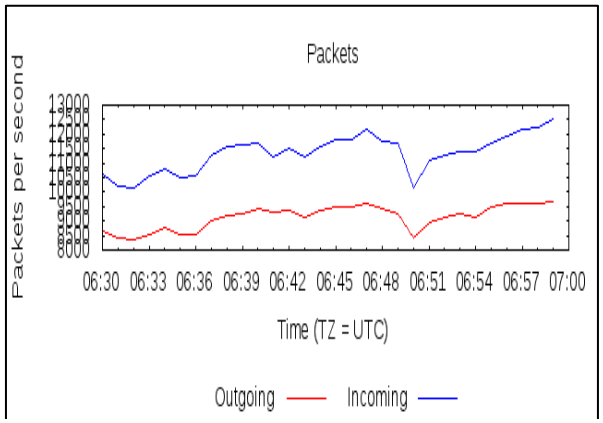

Figure 5 - the amount of input and output pocket in collection of data (Group)

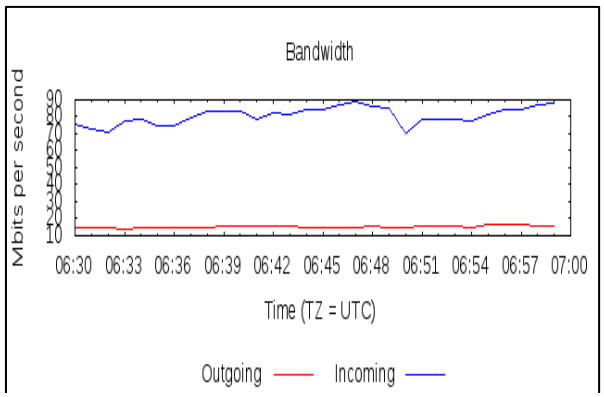

Figure 6 - the amount of input and output broadband in the collection of data (Group)

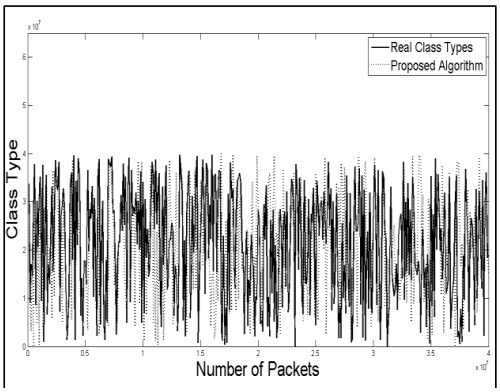

Figure 7 - the results of implementation algorithm represented on the collection of data (Group)

5 Conclusion

In this paper an algorithm has proposed to online identifying the type of input flow. This algorithm is based on applying learning machine techniques and classification. In order to present practical classifier we used main factors analysis method. The procedure of this process is teaching the classifier based on train data and it is used online during the algorithm implementation. Finally we used the proposed algorithm on some data sets of Waikato University then tested and implemented it, and found highly suitable accuracy.

Table 6 - represented algorithm accuracy for different collection of data

Data Set Number	Data Set Name	Accuracy (%)
1	ISPDSL-II/20100106-083000-0.dsl	81.23
2	ISPDSL-II/20100106-090000-0.dsl	74.2
3	ISPDSL-II/20100106-093000-0.dsl	76.1
4	ISPDSL-II/20100106-100000-0.dsl	78.14
5	ISPDSL-II/20100106-103000-0.dsl	81.1
6	ISPDSL-II/20100106-110000-0.dsl	73.48
7	ISPDSL-II/20100106-133000-0.dsl	80.26
8	ISPDSL-II/20100106-030946-0.dsl	81.78
9	ISPDSL-II/20100106-033000-0.dsl	79.12
10	ISPDSL-II/20100106-040000-0.dsl	71.26
11	ISPDSL-II/20100106-043000-0.dsl	74.16
12	ISPDSL-II/20100106-050000-0.dsl	83.24
13	ISPDSL-II/20100106-053000-0.dsl	75.48
14	ISPDSL-II/20100106-060000-0.dsl	72.19
15	ISPDSL-II/20100106-063000-0.dsl	79.14
16	ISPDSL-II/20100106-070000-0.dsl	81.3
17	ISPDSL-II/20100106-073000-0.dsl	74.97

Acknowledgment

Many thanks to my mother and my father, who show me the initial path of life, and always they have been my companion and supporter.

References

Al-Shaer, E. (2014). Classification and Discovery of Firewalls Policy Anomalies *Automated Firewall Analytics* (pp. 1-24): Springer.

Alcock, S., Lorier, P., & Nelson, R. (2012). Libtrace: a packet capture and analysis library. *ACM SIGCOMM Computer Communication Review, 42*(2), 42-48.

Antikainen, M., Aura, T., & Särelä, M. (2014). Denial-of-service attacks in Bloom-filter-based forwarding. *IEEE/ACM Transactions on Networking (TON), 22*(5), 1463-1476.

Bogdanoski, M., Suminoski, T., & Risteski, A. (2013). Analysis of the SYN Flood DoS Attack. *International Journal of Computer Network and Information Security (IJCNIS), 5*(8), 1-11.

Brownlee, N., Mills, C., & Ruth, G. (1999). Traffic flow measurement: Architecture. *Traffic*.

Brownlee, N., Mills, C., & Ruth, G. (1999). Traffic flow measurement: architecture (RFC 2722). *Outubro*.

Callado, A., Kamienski, C., Szabó, G., Gerö, B. P., Kelner, J., Fernandes, S., & Sadok, D. (2009). A survey on internet traffic identification. *Communications Surveys & Tutorials, IEEE, 11*(3), 37-52.

Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks, 32*, 333-338.

Darwish, M., Ouda, A., & Capretz, L. F. (2013). *Cloud-based DDoS attacks and defenses.* Paper presented at the Information Society (i-Society), 2013 International Conference on.

Eckhardt, J., Mühlbauer, T., AlTurki, M., Meseguer, J., & Wirsing, M. (2012). Stable availability under denial of service attacks through formal patterns *Fundamental Approaches to Software Engineering* (pp. 78-93): Springer.

Fiandrotti, A., Gaeta, R., & Grangetto, M. (2015). Simple Countermeasures to Mitigate the Effect of Pollution Attack in Network Coding-Based Peer-to-Peer Live Streaming. *Multimedia, IEEE Transactions on, 17*(4), 562-573.

Group, W. N. R. WITS: Waikato Internet Traffic Storage.

Hadi, A. D. A., Azmat, F. H., & Ali, F. H. M. (2013). *IDS Using Mitigation Rules Approach to Mitigate ICMP Attacks.* Paper presented at the Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on.

Jun, J.-H., Kim, M.-J., Cho, J.-H., Ahn, C.-W., & Kim, S.-H. (2014). Detection Method of Distributed Denial-of-Service Flooding Attacks Using Analysis of Flow Information. *The Journal of The Institute of Internet, Broadcasting and Communication, 14*(1), 203-209.

Kazantzidis, M., Gerla, M., & Lee, S. (2001). *RFC 3697: Permissible throughput network for adaptative multimedia in AODV MANETs.* Paper presented at the IEEE ICC 2001.

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review, 34*(2), 39-53.

Pelechrinis, K., Iliofotou, M., & Krishnamurthy, S. V. (2011). Denial of service attacks in wireless networks: The case of jammers. *Communications Surveys & Tutorials, IEEE, 13*(2), 245-257.

Rajahalme, J., Amante, S., Jiang, S., & Carpenter, B. (2011). IPv6 flow label specification.

Sheth, C., Thakker, R. A., Rahman, H., Abdullah, L., Joshi, R., Singh, M., . . . Vijayakumar, T. (2014). Performance Optimization of Network Firewalls by Rulebase Reordering based on Traffic Conditions. *International Journal Of Computer Science And Network Solutions*.

Timofte, R., Zimmermann, K., & Van Gool, L. (2014). Multi-view traffic sign detection, recognition, and 3d localisation. *Machine Vision and Applications, 25*(3), 633-647.

Van Raamsdonk, M. (2014). Evaporating firewalls. *Journal of High Energy Physics, 2014*(11), 1-16.

Yu, L., & Liu, H. (2003). *Feature selection for high-dimensional data: A fast correlation-based filter solution.* Paper presented at the ICML.

Zaklouta, F., & Stanciulescu, B. (2014). Real-time traffic sign recognition in three stages. *Robotics and autonomous systems, 62*(1), 16-24.