Journal of Computer Science

Original Research Paper

Detecting an Anomaly Behavior through Enhancing the Mechanism of Packet Filtering

Mohammed Nazeh Abdul Wahid and Azizol Abdullah

Department of Communication Technology and Networks,
Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia

Article history
Received: 12-02-2014
Revised: 15-10-2014
Accepted: 01-04-2015

Corresponding Author:
Mohammed Nazeh Abdul Wahid Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia
Email: Mohamed_ncs@yahoo.com

Abstract: The fundamental task of the Network Traffic Analysis is the ability of capturing and monitoring all the network traffics (incoming and outgoing) for local area network LAN and how the network analyzer is able to analyze and detect errors or any type of suspicious activities such as intruders. The idea of this research is to use flexible packet filtering to filter out the captured network traffics. The proposed packet inspection will isolate the captured traffics based on their source using traffic source separation 'TSS' strategy, during the separation operation the traffic signature will be examined with the stored signatures of the system database using Traffic Signature Matching. The experiment results shows that by using a User Profile Filter (UPF) that will be based on SVM and examining the traffic signature, the total of error received from the traffic classifier has been reduced to 0.5% and the traffic capturing speed has been increased in comparing with the standard methods of the traffic analyzers.

Keywords: Network Traffic Analysis, Packet Filtering, Anomaly Detection, User Profile Filter, Support Vector Machine, Traffic Signature Matching

Introduction

The Network Traffic Analysis is the tool that allows users to monitor and view the network traffics details. In this research we have explained in depth about the new generation of Network Traffic Analysis and its abilities to capture and monitor all types network traffics (incoming and outgoing) for local area network LAN. Our proposed method is actually depending on the network traffic analyzer to capture and analyze the network traffics and we have proposed a special technique that detects anomalies while monitoring network traffics, this technique is called the flexible packet filtering of the support vector machine. So, in this research we have merged the analyzed results for both of the flexible packet filtering and the support vector machine algorithm that we used to get the better classification of the captured network traffics and to detect anomalies. This method will help improving the abilities of the network analyzer to determine whether the captured traffics are normal or have an anomaly behavior with an expected of a very low rate of false alarm. The purpose of combining these methods is to save time, employees and effort of monitoring the traffics and handling the alarm that indicates to the presence of attack. These points are very important and taking the consideration of the researchers especially those are trying to overcome the weaknesses that degrading the performance of the network traffic analyzer such as time of fixing a problem that might be not existed due to the high rate of false alarm in most of network analyzers (Allen *et al*., 2005). So, we have come out with a comprehensive application that can adept the changes of the network behavior and able to handle more than one function simultaneously. Using this application would help us avoiding the attackers and preventing them to penetrate our network environment and this could also reflect the advantages of saving time, money and working staff. What polis saying every time is that "we did not catch all the criminals but we know their victims"? So let us consider that the criminal is the attackers with their IP-addresses and the victims are the network. Unfortunately, it is not possible to reliably determine the source of a received IP packet, as the protocol does not provide authentication of the packet based on the source address field, which can be easily faked (IP Spoofing). Furthermore, also the Internet routing infrastructure does not keep useful information about forwarded packets. In this research, anomaly detection is used in checking the IP Header and TCP or UDP Header using the payload and

Science Publications

traffic four metrics (Belenky and Ansari, 2007; Mahoney, 2003), our strategy is based on viewing the header of the packet using packet decode and the observation that fundamentally all anomaly detection methods must be first defined as normal behavior; anomalies then become the deviations from this baseline behavior. Most anomaly detection algorithms require a set of purely normal data to train the model and they implicitly assume that anomalies can be treated as patterns not observed before. Also most anomaly detection methods are limited to analyze the entire traffic as one entity, which makes them unable to quantify network anomalies and their validities are affected when many anomalous activities occur simultaneously (John and Tafvelin, 2007).

Many researchers have proposed different techniques and methodologies for capturing and recognizing anomalies, some uses the IDS and test it using the DARPA 99 dataset (DAPRAdataset) and others used the standard CISCO net-flow analyzer (CISCO NetFlow) with their own dataset. Recently researchers have turned towered the use of the Support Vector Machine to recognize anomaly behavior (Kim *et al*., 2005; Tran *et al*., 2004) or unknown attacks under different circumstances and conditions. Most of these techniques works correctly and gives good results. Our new proposed packet inspection that has been deployed to capture an anomaly behaviors is relying on support vector machine algorithm that identify the new traffic behavior by comparing it with the stored user profile and helps to flag data whether normal or anomaly and returned data will be added to the oldest pattern and constructs new updated normal profile that contain more information about normal user behavior. In SVM, if we define the distance from the separating hyper-plan to the nearest expression vector as the margin of the hyper-plane (Kim *et al*., 2005), then the SVM algorithm selects the maximum margin separating hyper-plane. Selecting these particular hyper-planes maximizes the SVM s ability to predict the correct classification of unseen pattern. Other researches in supervised anomaly detection can be considered as performing generative modeling. These approaches attempt to build some kind of a model over the normal data and then check to see how well new data fits into that model. An approach for modeling normal sequences using look ahead pairs and contiguous sequences has been also consider revising in this study. A statistical method for ranking each sequence by comparing how often the sequence is known to occur in normal traces with how often it is expected to occur in intrusions have some mathematical defects and difficulties to achieve (Jung *et al*., 2004). The libpcap tool has greatly simplified the task of acquiring network packets for measurement. The limitation of the tool is its inability to analyze the captured data, it will only capture the data and the programmer or network administrator is left to carry out analysis manually. This task can be time consuming and cumbersome and in most cases accurate information about the network is not obtained. Some researchers have developed modular software architectures for extensible system, however only a few of these systems are optimized to handle large amount of data and continuous monitoring. Other researchers have developed systems for streaming data through protocol layers and routing functions, but not much attention has been given to the analysis of large/huge or broad data collected over time. One approach uses a prediction model obtained by training decision trees over normal data, while others use neural networks to obtain the model or non-stationary models to detect novel attacks, but still facing a high rate of false alarm. Lane and Bradley (2008) performed anomaly detection on unlabeled data by looking at user profiles and comparing the activity during an intrusion to the activity during normal use. Similar approach of creating user profiles using semi-incremental techniques has also proposed by other researchers. Barbara used pseudo-Bayes estimators to enhance detection of novel attacks while reducing the false alarm rate as much as possible. A technique developed at SRI in the EMERALD system uses historical records as its normal training data. It then compares distribution of a new data to the distribution obtained from those historical records and then, the differences between the distributions indicate an intrusion. Recent works estimate parameters of a probabilistic model over the normal data and compute how well new data fits into the model. Kim *et al.* (2005; Lane and Bradley, 2008), traffic packets are projected to four matrices according to different bytes of the IP address and then an abnormality detection method for large scale network is proposed. Besides analyzing the characteristic of the IP address, many researchers try to discover the statistical character of users' behaviors and perform abnormal behavior detection. The protocol, client, server port, total data transferred are used to describe the users' communication patterns and to cluster them into different Community Of Interests (COI). Through analyzing the characteristics of the COIs, many abnormal behavior detection methods are designed. The abnormal behaviors can be detected by analyzing the protocols, packets size and flow size. In implementing this idea, it is usually required to check every packet to get the detailed address information. In actual application, this may affect the efficiency of the real-time traffic monitoring. To avoid this and improve the efficiency, some researchers analyze the statistics of the traffic packets (total number of bytes, total number of packets) and successfully propose many schemes to discover the anomalies only when traffic pattern changes due to attacks such as DDOS for (Mahoney, 2003; Giorgi and Narduzzi, 2008). Giorgi and Narduzzi (2008;

Mahoney, 2003; Chandalia and Rish, 2007; Qin *et al*., 2009; Somayaji and Forrest, 2000; Srinoy, 2007) in data profiling, researchers have developed a mechanism for maintaining an updated user profile. To test their mechanism, they collected 15,000 UNIX commands for each of 70 users and decomposed them into 150 blocks of 100 commands. The data set (which we will refer to as the SEA data) was tested by choosing 50 users to serve as intrusion targets and designating the remaining 20 users as masqueraders by mixing their data in with that of the other 50 users. They replaced the training data set by the most recent 5000 commands after every 2000 test commands for a given user. The previous training data remained in place in the event that alarms were raised so that anomalous data would not overwrite the existing training data. They applied a number of different approaches (such as one-step Markov, IPAM, sequence matching and compression) to the SEA dataset. However, some of their tests showed an error rate of as much as 65%, since they simply updated normal profiles every 2000 test commands without considering the variety of pattern change made by the user. So, our new proposed method that filter traffics based on several characteristics is aiming to make the user profile filter fits with the algorithm of support vector machine, from here we see the need of developing an efficient and effective network traffic prediction and monitoring methods for detecting anomalies and this is mainly complicated by the following factors: (a) Users may slowly change their behavior with the system and time evolution (e.g., the traffic in a network may present changes and variations) and therefore, any associated algorithm should be capable of dynamically adapting to these changes and evolutions to avoid the high rate of false alarm and the effort spent trying to identify the point that leads to trigger the alarm. (b) The analysis and prediction about the expected traffic should be based on normal data (filtered and reliable data), while all the data that may be noise/anomalies and may affect the accuracy and correctness of the normal behavior prediction should be excluded, these problems must be solved in order to apply the User Profile Filter that will help us detecting network anomalies while reducing the false alarm rate. (c) The fundamental difficulties in achieving an efficient and accurate declaration of an intrusion and how to train these data with support vector machine algorithm that suppose to classify the traffic into normal or attack.

Materials and Methods

Our method works based on monitoring the four predefined traffic metrics as its shown in (Table 1) that capture the flow statistics of the network along with the stored user profile in the system database. In order to prove the power of the new method, we did build an application that detects network anomalies using the proposed method. The result of the experiments proves that by using the four simple metrics from the flow data and by monitoring the user's activities, we do not only effectively detect attacks but can also identify the network traffic anomalies. Then, the packet decode will decrypts the traffic and view its header and payload, this will allow zooming in on a specific details which will associate the comparison of the user activities. Internet traffic measurement in (Giorgi and Narduzzi, 2008) is essential for monitoring trends, network planning and anomaly traffic detection. In general, simple packet- or byte-counting methods with SNMP have been widely used for easy and useful network administration (Mahoney, 2003). In addition, the passive traffic measurement approach that collects and analyzes packets at routers side or dedicated machines is also popular. However, traffic measurement will be more difficult in the next-generation Internet with the features of high-speed links or new protocols such as IPv6 or MIPv6. In Table 1 are some examples of anomaly attacks those can be captured and identified using the four predefined traffic metrics.

The most important part of this research is the traffic filtering. First, to sniff network traffics we used packet sniffer which is followed by packet decode to capture and reveal traffic header and payload. This has been done using flexible packet filtering which is combination of both the static and dynamic packet filtering. WinPcap (traffic library) will be also used to determine the traffic validity. After traffics have been captured, the data packets will be isolated based on their source using traffic source separation 'TSS' strategy and during the separation operation the traffic signature will be examined with the stored signatures of the system database using Traffic Signature Matching. According to the proposed technique, another filter applied which is 'User Profile Filter' (UPF) that will be based on SVM that have the record of the normal user activities and ability of classifying network traffic into normal or attack.

Table 1. Anomaly attacks those can be captured using traffic four metrics

Traffic four metrics				
Over types of attack	Total Byte	Total packet	D-socket	D-port
Flooding	High	High	Normal	Normal
TCP SYN	Normal	Normal	High	Normal
Port-Scan	Normal	Normal	Normal	High

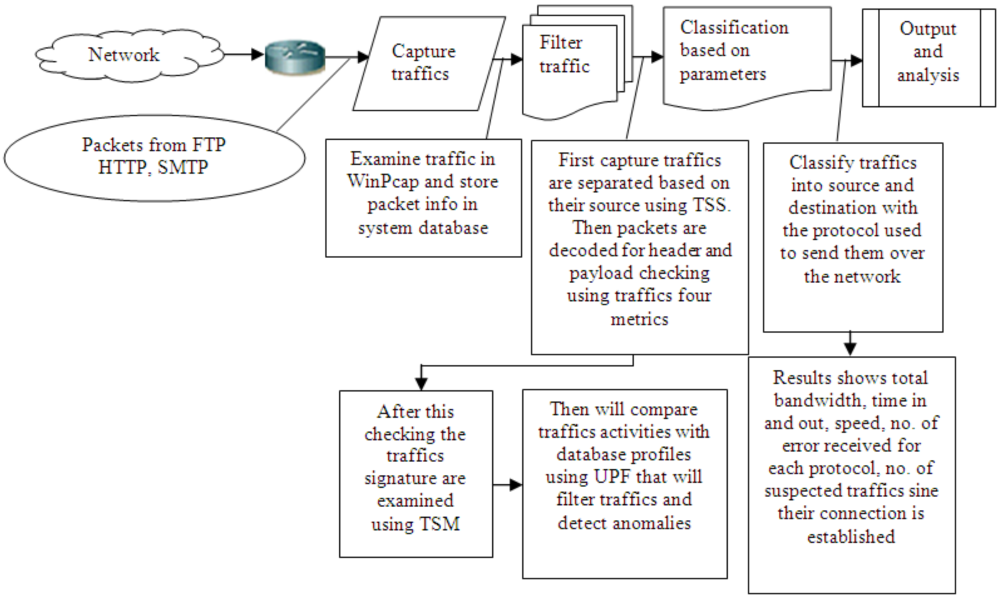

Network
Packets from FTP HTTP, SMTP
Capture traffics
Filter traffic
Classification based on parameters
Output and analysis
Examine traffic in WinPcap and store packet info in system database
First capture traffics are separated based on their source using TSS. Then packets are decoded for header and payload checking using traffics four metrics
Classify traffics into source and destination with the protocol used to send them over the network
After this checking the traffics signature are examined using TSM
Then will compare traffics activities with database profiles using UPF that will filter traffics and detect anomalies
Results shows total bandwidth, time in and out, speed, no. of error received for each protocol, no. of suspected traffics sine their connection is established

Fig.1. System design

The use of Packet Decode is to get information that could help analyzing traffics in depth and understand how it has been sent and what the destination that the data should be delivered to by viewing its header and payload using specific parameters in each step. The parameters in this system are depends on each other, because from the above system design (Fig. 1) we can see that each step is relying on the previous one. So, in case of failure in one of the steps, the next step will not be able to analyze traffics correctly. So, the consequences will be failed to decode or to determine whether the packet is clear or not. By using the Smart Sniff we will start capture the traffics. So the first step for the system design is:

Capture Traffics: This process starts with packet sniffer to capture the traffics from the internet and this function will allow monitoring and capturing all the traffics in Local Area Network (LAN). So, it could be in the university or in star-bucks or home or in anywhere that can find internet connection. In this step, the proposed analyzer is linked with WinPcap which is internet packet library, that would improve the ability of network analyzer to predict the traffics that are going to be captured and examining the packets with stored database, so the packet data are suppose to be matched with the stored WinPcap database to determine the packet type and the protocol used to send it, it could be (TCP or UDP, ICMP, IGMP). When the traffics are captured, there is only one gate to bypass through our adapter. This gate is a WinPcap. If the packet does not match a WinPcap database, the system will indicate it as an infected packet. So once it finishes the process of checking packets, it will forward the information to the next step.

Filter Traffics: all the captured traffics are a WinPcap database. But, the traffics can be forged or modified to act as normal. So, while capturing the traffics, the data packets will be posed to be filtered using the proposed method. At this stage, the system will use the packets decode to reveal the packet headers and payload and the Flexible packet filtering over Support Vector Machine shall determine the traffic reliability.

Traffic Signature Matching

At this stage as it is shown in the system design the signatures of the received IP packets reply will be checked using Traffic Signature Matching which known also by misuse detection that will be responsible to examine the captured traffics signatures and alert if one of the captured traffic signature has matched the stored signatures. In this study we depend on the traffic four metrics to identify the type of attacks for both the assumed signature and for new attacks that caused by anomalies.

User Profile Filter

While the traffic signature has been examined and data packet been forwarded to be checked based on anomaly behavior. In order to do so, we need to perform anomaly detection on unlabeled data by looking at the user profile and comparing the activities during an intrusion to the activities during normal use and then every user profile will be recorded in the system database in the server side. So, we need to create the user profile filter that will be the responsible to filter out and comparing the captured data with the stored profiles depending on the records of the normal user behavior as shown in (Fig. 2) to detect any effective deviations to enhance the detection of novel attacks while reducing the false alarm rate, this can be done using SVM with its maximize margin that will be able to recognize anomaly behavior.

The Structure Design of SVM

The support vector machine algorithm creates normal profile of the users and helps to flag data whether normal or anomaly; the returned data will be added to the oldest pattern and constructs new updated normal profile that contain more information about normal user behavior. Since it is very difficult to set any predefined rules for identifying correctly attack traffics and we have to consider that there is no major difference between normal and attack traffics. In the SVM, if we define the distance from the separating hyper-plan to the nearest expression vector as the margin of the hyper-plane, then, the support vector selects the maximum margin separating the hyper-plane. Selecting this particular hyper-plane maximizes the SVM s ability to predict the correct classification of unseen pattern. In such away this tool will help us detect and identify anomalies while monitoring the network traffics as well as reducing the rate of false alarm. Figure 2 shows how the SVM works using our method to flag data whether normal or attack and the space which is between them has been left to the developer to decide the burden of the false alarm rate.

So, Data profiling is an analysis of the candidate data sources for a data warehouse to clarify the structure, content, relationships and derivation rules of the data (Wikipedia, 2013). Profiling helps to understand anomalies and to assess data quality, but also to discover, register and assess the enterprises metadata (Beauquier and Hu, 2007). Thus the purpose of data profiling is both to validate metadata when it is available and to discover metadata when it is not (Kim *et al*., 2005). The result of the experiment analysis used both strategies, to determine suitability of the candidate source systems and give the basis for an early go/no-go decision and tactically, to identify and report problems for later solution design and to level sponsors' expectations. The Fig. 3 shows how the support vector machine algorithm works to create an efficient and reliable user profile that will associate the packet inspection to determine the traffic behavior.

What the most researchers face with support vector machine is the lake of the data noise, Selecting an appropriate set of features for normal profiles is directly related to achieving efficient anomaly detection systems since data dimensionality can be reduced; however, even though the most concise set of features has been selected, the amount of data will be still increasing when the most current trends are continuously added to the normal profile. Figure 3, explaining how the captured data are transferred to be examined with the record of the system database using the maximize margin of the support vector machine to create a reliable user profile. This algorithm is able for adapting for any changes to the users behavior while flagging anomaly when the changes of the user's behavior have exceeds the limit. Figure 3, shows the hyper-plane of the maximize margin that will be responsible for matching the received data packet with stored data for certain user after excluding the noise data. The theory example of SVM; we have L training points, where each input xi has D attributes (i.e., is of dimensionality D) and is in one of two classes' y_i = -1 or +1, i.e., our training data is of the form:

$$\{Xi, yi\}\, where\, i = 1...L, yi \in \{-1,1\}, X \in \Re^{D}$$

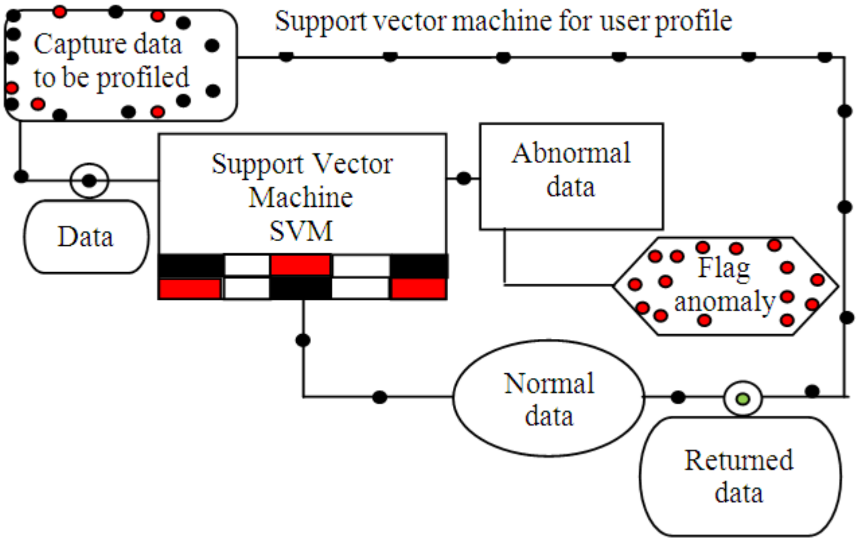

Support vector machine for user profile
Capture data to be profiled
Support Vector Machine SVM
Abnormal data
Data
Flag anomaly
Normal data
Returned data

Fig. 2. The structure design of user profile

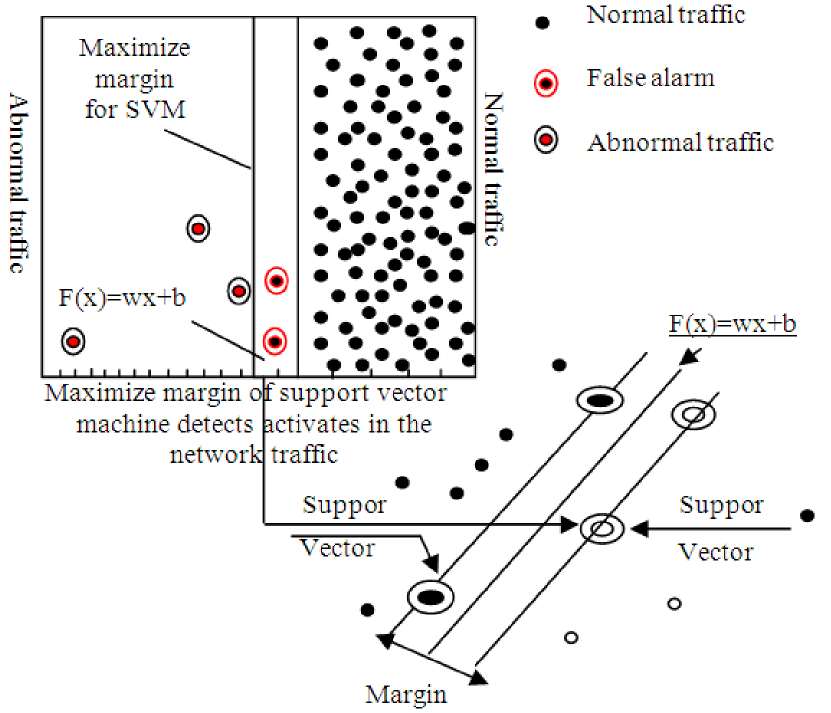


Fig. 3. The margin of SVM classifying traffics into normal and attack

Here we assume the data is linearly separable, meaning that we can draw a line on a graph of X_1 Vs. X_2 separating the two predefined classes when D = 2 and a hyper-plane on graphs of X_1; X_2 : : : X_D for when D>2. This hyper-plane can be described by w. x + b = 0 where: w is normal to the hyper-plane, b over ||w|| is the perpendicular distance from the hyper-plane to the origin. So, the final form of the SVM that we used is: f (x) = w x + b. Referring to Fig. 2, implementing a SVM boils down to selecting the variables w and b so that our training data can be described by closest to the separating hyper-plane, i.e., the Support Vectors (shown in circles in the diagram), then the two planes H1 and H2 that these points lie on can be described by:

$$
\begin{array}{ll}
x_i.w+b \geq +1 & for\, y_i = +1 \\
x_i.w+b \leq -1 & for\, y_i = -1 \\
x_i.w+b = +1 & for\, H_1 \\
x_i.w+b = -1 & for\, H_2
\end{array}
$$

These two equations can be combined into:

$$y_i(x_i.w+b)-1 \geq 0\forall_i$$

Some researchers have tried to use clustering algorithms in order to alleviate the burden of the data dimensionality. Nonetheless, their approaches also were not free from data contamination due to irrelevant information ("noise"). Furthermore, when an existing profile is completely replaced by a new profile, the noise effect worsens. We address this problem by applying a Support Vector Machine (SVM), a mathematical learning algorithm was designed for classification purposes. The main idea behind SVM algorithms is that classification of data can be done by constructing an imaginary hyper-plane in a multidimensional feature space that separates different classes. SVMs classify data by determining a small subset of the training data, called the Support Vectors (SVs), where support vectors draw a hyper-plane in feature space in order to find a decision rule with good generalization ability. The noise effect can be minimized by discovering the hyper plane that maximizes the margin between two classes in the feature space using SVMs. One more problem has been solved in this research with the Binary classification for data that is not fully linearly separable, In order to extend the SVM methodology to handle data that is not fully linearly separable, we relax the constraints of the previous hyper plane margin slightly to allow for misclassified points. This is done by introducing a new positive slack variable ξ_i, i = 1,…L: which can be described by:

$$
\begin{array}{ll}
x_i.w+b \geq +1-\xi i & for\;\; y_i = +1 \\
x_i.w+b \leq +1+\xi i & for\;\; y_i = -1 \\
\xi i \geq 0\forall_i &
\end{array}
$$

This can be combined using this formula:

$$y_i\,(x_i\,.w+b)-1+\xi_i\ \geq 0\ where\ \xi_i\ \geq 0\forall_i$$

By applying this equation we get a perfect soft margin SVM, data points on the incorrect side of the margin boundary have a penalty that increases with the distance from it. As we are trying internally to reduce the number of misclassifications, we have to consider also the regression of a new data and the aggression of unseen data. So, Instead of attempting to classify new unseen variables x” into one of two categories y’ = ±1, we now wish to predict a real-valued output for y’ so that our training data is of the form:

$$\{X_i\,,y_i\}\,where\,i=1...L,y_i\ \in\Re,x\in\Re^D$$

So, by using these equations, the problem that has been found in the hyper plane of the SVM with packet inspection is solved and the result gives a very low rate of misclassified network traffics.

Classification Based on Parameters: The output of the traffics will be classified using several parameters, such as parameters those are responsible to calculate the traffics bandwidth, total packets received, total error received and time in and out. All these parameters help in understanding the traffics details and behavior, so, users can troubleshoot any problem they might face while trying maintain the connections between the parties. The classification process is also responsible to classify the traffics into source, destination and the protocol used to send them over the network. After that, the information about the traffics will be forwarded to the final process.

Results and Analysis: The results shows the bandwidth rates for each protocol based on its usage in the network and the total packet error of TCP connections and view the results in graph to make a reasonable comparison between the protocols. The output traffics are going to be presented on an organized interface to make the users able to understand the traffic details. This software would operate perfectly in Domain Naming System (DNS) which have authentications for their staff members or group and all users are posses to work under same security concept and roles to avoid and internal attackers and to eliminate the false alarm of our known users.

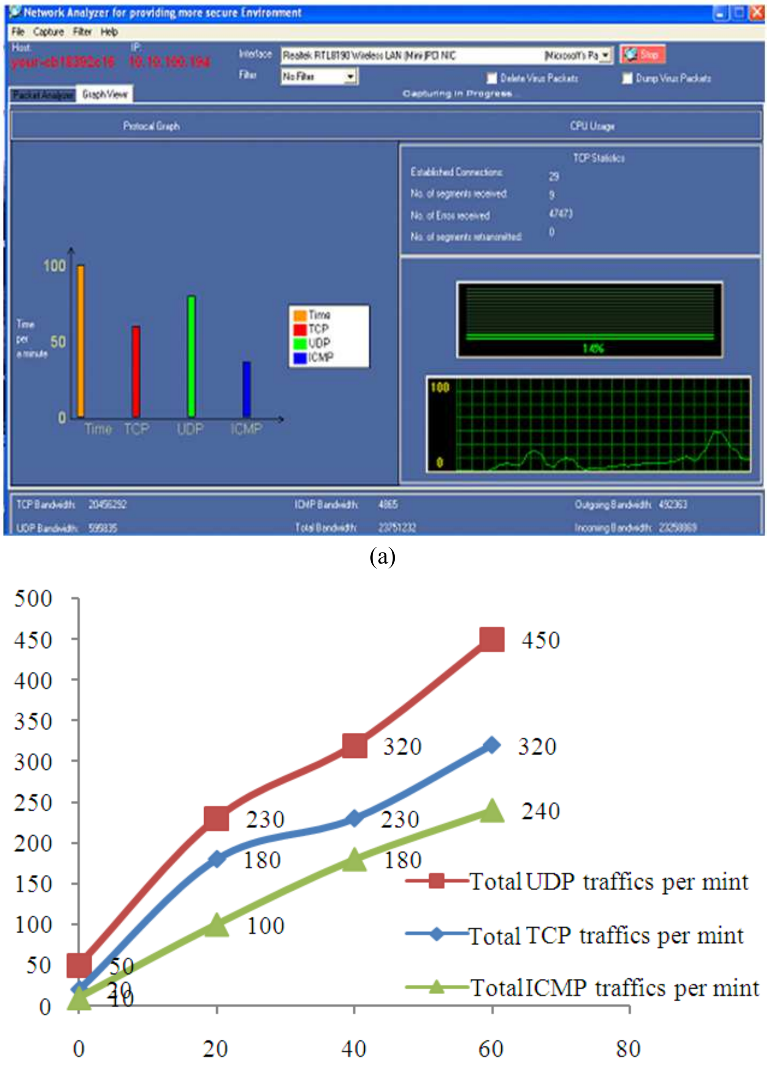
Experiment Results

Many researchers has proposed different techniques to capture and recognize anomalies, some uses the IDS and the testing has been done using the DARPA 99 dataset and others used the standard CISCO net-flow analyzer with their own dataset. Most of the network analyzer depends on real dataset and work on real environments. So, in these cases the testing is effected by the environment that the application is running with. Our first test has been done in the faculty of computer science (UPM), using the standard method that we have followed earlier to develop the software such as method used in many nowadays researches of network analyzers and we have captured a hug number of traffics with a proportion of 12% of error received, these errors might be misclassified traffics or it could be unknown attackers or private traffics (Block-Traffic) that the network analyzer has no authentication to filter. Our latest test has been done in the same environment but this time using our proposed new method that filter traffics using flexible packet filtering and separating the captured traffics based on their source to enhance the classification operation and using the support vector machine algorithm to recognize anomaly attacks and constructing a reliable user profile.

Our system results appear as the following; almost all types of traffics have been captured and the traffic filtering speed over the number of captured traffics has been increased up to 15% per mint comparing to our previous method that has been used in the majority of network analyzers. The identification of the network traffic protocols those are traversing over the network have been improved and the result has comes with zero percent of misclassified traffic as it’s shown in Fig. (4a) except for those who are blocked by their private network admin. Also the traffic signature matching which is known also by misuse detection, gives an excellent results by alerting users to the presence of viruses and hacker attacks those who are typically generate a recognizable pattern or “signature” of packets. The Fig. 4a and b shows the captured traffics rate per mint with the total bandwidth of each protocol presented in the system interface.

Figure 4a showing also the rate of the captured traffics based on the protocols used to send them over the network and the measurement has been done according to the number of captured traffics and bandwidth rate for each protocol per mint to show up the variables that we have got from the method and how did we refer to them through the comparison. Figure (4b) presenting the number of total captured traffics that is almost over 1000 traffics, in this test, UDP has higher traffics rate than TCP and ICMP, for that particular mint whether the higher bandwidth rate was for TCP followed by UDP then ICMP as it is shown in Fig. (4a) in this section.

The Fig. 5 showing us the total TCP bandwidth over the number of received errors. First test has been done using the standard method that most of the network analyzers follow to capture and analyze the data traffic and the second test was measured using our new proposed method.



Network Analyzer for providing more secure Environment

Host: your-cb18392c16 IP: 10.10.100.194

Interface: Realtek RTL8190 Wireless LAN (Mini PCI) NIC (Microsoft's Pa

Filter: No Filter

Capturing in Progress

TCP Statistics

Established Connections: 29

No. of segments received: 9

No. of Erros received: 47473

No. of segments retransmitted: 0

TCP Bandwidth: 20456292

UDP Bandwidth: 599835

ICMP Bandwidth: 4865

Total Bandwidth: 23751232

Outgoing Bandwidth: 492363

Incoming Bandwidth: 23258869

(a)

500
450
400
350
300
250
200
150
100
50
0
0
20
40
60
80
450
320
320
230
230
240
180
180
100
50
20
10
Total UDP traffics per mint
Total TCP traffics per mint
Total ICMP traffics per mint

(b)

Fig. 4a and b shows the captured protocols rate per mint measured by the total bandwidth for each protocol

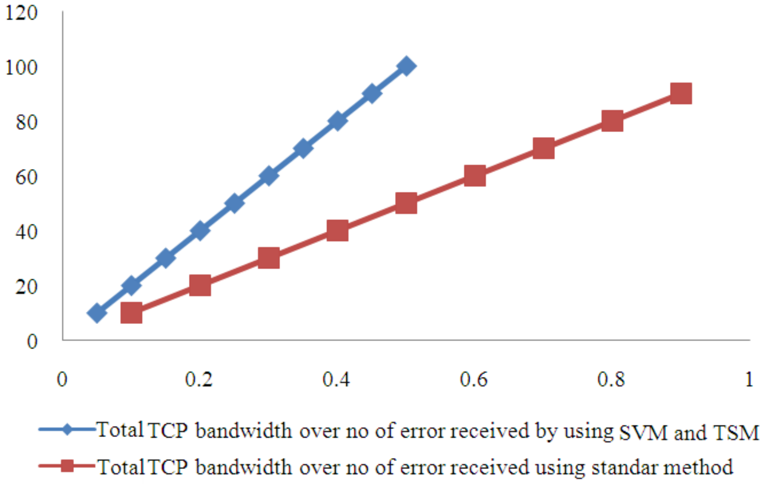


Fig. 5. Shows the total of TCP bandwidth over number of error received by both the standard method and the proposed method

The results shows that by using flexible packet filtering and classifying the captured network traffics using the SVM and examining the traffic signature the total of error received has been reduced to 0.5%, whereas; the total of error received using the standard method reach's to 1.3% of total packet error and the testing for both methods has been done under same circumstances and same environment and running time.

Discussion

The network IDS devices use passive network monitoring extensively to detect possible threats. Through passive monitoring, the security admin can gain a thorough understanding of the network's topology: What services are available, what operating systems are in use and what vulnerabilities may be exposed on the network. Much of this data can be gathered in an automated, non-intrusive manner through the use of standard tools, While the concepts presented here are not difficult to understand, the reader should have at-least an intermediate understanding of IP and a base-level familiarity with operation of network sniffers. As we mentioned earlier that our proposed method is depending on the network traffic monitoring to capture and analyze the network traffics and using a special technique that detects anomalies while monitoring the network traffics called flexible packet filtering of the support vector machine. In this research we have merged the analyzed results for both of the flexible packet filtering and the support vector machine to get the best classification for the captured traffics and to detect anomalies. The purpose is to save time, employees and effort of monitoring the traffics and handling the alarm that indicates the presence of attack.

Conclusion

We concluded that by using the SVM alone to classify and detect anomalies traffics do not give very good results as network features are used for learning without processing. But by using the network traffic prediction technique to analyze and detect an anomaly behaviors and by applying the Flexible Packet Filtering that will be supported by Traffic Signature Matching including the Traffic source separation technique, the result shows that SVM works perfectly and gives better results than working alone with a proportion of 0.5% of misclassified traffics with the lowest false alarm rate which not exceeds 3% of total filtered traffics. Using the User profile filter our system will be able to identify and detect an anomaly behaviors and trace them back to their original source, the tracing is one of the major problem that we will consider it in our future work, because it is not possible to reliably determine the source of a received IP packet, as the protocol does not provide authentication of the packet based on the source address field, which can be easily faked (IP Spoofing). Furthermore the Internet routing infrastructure also does not keep information about forwarded packets. So, our system will be able to detect and identify all types of network attacks and an alert will be triggered indicating to their occurrence, after that it will classify the captured traffics into source, destination, port number and the protocols used to send them over the network.

Acknowledgement

I would like to acknowledge and appreciate the full support that has been given to me from University Putra Malaysia and especially my supervisory committee.

Author’s Contributions

Mohammed Nazeh Abdul Wahid: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Dr. Azizol Abdullah: Participated in results testing and data-analysis.

Ethics

The author is responsible for all the materials and the methodology involved in this publication and it is a copyright of journal of computer science.

References

Allen, W.H., G.A. Marin and L.A. Rivera, 2005. Automated detection of malicious reconnaissance to enhance network security. Proceedings of the SoutheastCon, Apr. 8-10, IEEE Xplore Press, pp: 450-454. DOI: 10.1109/SECON.2005.1423286

Beauquier, J. and Y. Hu, 2007. Intrusion detection based on distance combination. Int. J. Comput. Electrical, Automat. Control Inform. Eng., 1: 1932-1940.

Belenky, A. and N. Ansari, 2007. On deterministic packet marking. Comput. Netw., 51: 2677-2700. DOI: 10.1016/j.comnet.2006.11.020

Chandalia, G. and I. Rish, 2007. Blind source separation approach to performance diagnosis and dependency discovery. Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, Oct. 23-26, ACM, San Diego, pp: 259-264. DOI: 10.1145/1298306.1298342

CISCO NetFlow. Network Analyzer At www.cisco.com/warp/public/732/Tech/netflow/

DAPRAdataset. /http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/detections_1999.htmlS

Giorgi, G. and C. Narduzzi, 2008. Detection of anomalous behaviors in networks from traffic measurements. IEEE Trans. Instrumentation Measure., 57: 2782-2791. DOI: 10.1109/TIM.2008.926046

John, W. and S. Tafvelin, 2007. Analysis of internet backbone traffic and header anomalies observed. Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, Oct. 23-26, ACM, San Diego, pp: 111-116. DOI: 10.1145/1298306.1298321

Jung, J., V. Paxson, A.W. Berger and H. Balakrishnan, 2004. Fast portscan detection using sequential hypothesis testing. Proceedings of the IEEE Symposium on Security and Privacy, May 9-12, IEEE Xplore Press, pp: 211-225. DOI: 10.1109/SECPRI.2004.1301325

Kim, D.S., H.N. Nguyen and J.S. Park, 2005. Genetic algorithm to improve SVM based network intrusion detection system. Proceedings of the19th International Conference on Advanced Information Networking and Applications, Mar. 28-30, IEEE Xplore Press, pp: 155-158. DOI: 10.1109/AINA.2005.191

Lane and Bradley, 2008. Cost-sensitive modeling for fraud and intrusion detection 2008. Data mining based on intrusion detection system.

Mahoney, M.V., 2003. Network traffic anomaly detection based on packet bytes. Proceedings of the 2003 ACM Symposium on Applied Computing, Mar. 9-12, ACM, Melbourne, pp: 346-350. DOI: 10.1145/952532.952601

Qin, T., X. Guan, W. Li and P. Wang, 2009. Monitoring abnormal traffic flows based on independent component analysis. Proceedings of the International Conference on Communications, Jun. 14-18, IEEE Xplore Press, Dresden, pp: 1-5. DOI: 10.1109/ICC.2009.5199196

Somayaji, A. and S. Forrest, 2000. Automated response using system-call delays. Proceedings of the 9th USENIX Security Symposium, pp: 185.

Srinoy, S., 2007. Intrusion detection model based on particle swarm optimization and support vector machine. Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications, IEEE Xplore Press, Honolulu, pp: 186-192. DOI: 10.1109/CISDA.2007.368152

Tran, Q.A., H. Duan and X. Li, 2004. One-class support vector machine for anomaly network traffic detection. Proceedings of the 2nd Network Research Workshop of the 18th APAN, (APAN' 04), Cairns, Australia.