Journal of Computer Science 7 (5): 790-799, 2011
ISSN 1549-3636

Applying Packet Generator for Secure Network Environment

Mohammed N. Abdul Wahid and Zuriati Ahmed Zulkarnain
Department of Communication Technology and Networks,
Faculty of Computer Science and Information,
Technology University Putra Malaysia, Malaysia

Abstract: Problem statement: Viruses and hacker attacks typically generate a recognizable pattern or "signature" of packets. Most of Network Traffic Analyzer can identify these packets and alert the administrator to their presence on the network via email or page. **Approach:** Most traffics analyzers let you set alarms to be triggered when a particular pattern is seen. **Results:** Some network traffic analyzers can be programmed to send an email or page when these conditions are met. Of course; this assumes that the virus and its signature have been seen before and incorporated the analyzer's list of packet filters. ((The packet filters once started the filtering process and also by using packet decode together they can determine the traffic type whether it has normal or abnormal activities. **Conclusion/Recommendations:** In this study we used Packet Generator to generate a traffic that supposes to act the intruder or hacker signature to prove up that Network Traffic Analysis has the ability to detect like this kind of traffics. And also we have explained in depth about network traffic analysis and its ability to monitor all the network traffics (incoming and outgoing) and view their headers and payload and all other information such as traffic source and destination)).

Key words: Network traffic analysis, network security, packet generator, virus, attacks signatures, packets decode, filter traffics, security and forensics

INTRODUCTION

This research is a web-based network traffic analyzer using to generate instant reports on network traffic and users. And accurate details on data analysis, data interpretation and graphical presentation of results to correlates them and generates graphs and reports that help in understanding and troubleshooting network traffic. It allows you to monitor bandwidth and traffics in an interface specific level. The selectable graph allows you to zoom in on the filtering traffic, also shows the data points, which gives the traffic IN and traffic OUT details such as speed, volume, packets and utilization off the total bandwidth. Not only can you view the analysis network traffic reports, you can also custom select the time period for which you want to view what time. The rapid growth of the Internet in size, complexity and traffic types has made network management a challenging task (Aggarwal *et al*., 2003; Ashfaq *et al*., 2008; Babcock *et al*., 2003; Chen *et al*., 2004; Bon, 2009; Tan and Sherwood, 2006). The ability of a monitoring system to provide accurate information about the nature and type of the network traffic cannot be over emphasized. Information about who is generating the most traffic, what protocols are in use, where is the traffic originating from or where is the destination of the traffic can be very important to solving congestion problems. Many network administrators spend a lot of time, trying, to know what is degrading the performance of their network (Almulhem and Traore, 2005a; 2005b; Almulhem, 2009; Anaya *et al*., 2009; Riech and Laskov, 2006; Wang and Thomas, 2008).

A typical solution to congestion problem is to upgrade network infrastructure, i.e., replace servers with high end servers and increase the bandwidth. This solution is expensive, short term and does not scale. As soon as the upgrade is done the congestion problem will improve and after that we will gradually start suffering the spoil as the users change their behavior in response to the upgrade. The alternative solution to this problem is to deploy a scalable network traffic monitoring and analysis system, in order to understand the dynamics of the traffic and changes in the internet from time to time as well as the overall stability of the network (Datar *et al*., 2002; Domingos and Hulten, 2000; Ganti *et al*., 2002; O'Callaghan *et al*., 2002; SANS, 2007; Babcock *et al*., 2003). So, by monitoring network activities we

Corresponding Author: Mohammed N. Abdul Wahid, Department of Communication Technology and Networks, Faculty of Computer Science and Information, Technology University Putra Malaysia, Malaysia
Tel: 0060146401642

can also have advantages from detecting Denial of Service (DoS) attacks and the expert of bandwidth theft attacks. In order to conduct analysis of wide range of network behaviors, it is necessary to collect network traffic on a continuous basis rather than as a onetime event which only captures transient behaviors that provides insight into network problems. Collecting long term network traffic data will provide valuable information for improving and understanding the actual network dynamics. Most of the network traffic analysis systems are able to capture traffics from any kind of network, in order to identify the networks traffics patterns and protocols. These traffics are subdivided into two types:

- The normal traffics: these kinds of traffics that have normal activities are the most coming traffics from the Network
- The abnormal traffics: These kinds of traffics are seldom when we detect from the network because of the existence of the firewall or IDS. But some of them are able to penetrate the firewall or any other security devices such as IDS, IPS and so on

((Most of the current Network Traffic Analysis once they detect a suspicious activity or abnormal traffics they have nothing to do with unless they can view the abnormal traffics with red color indication to recognize it)). But in such a hug establishments or institutes who can afford to bay an original version of WinPcap that allows them to modify the infected packet to make it valid or if it is not modifiable they can drop or terminate it from the network. To overcome this weakness which has found in the previous system such as when they detect a suspicious activity or abnormal traffics they cannot do anything for that packet.

So, by our turn we ((aim to improve the Network Traffic Analysis system that we developed using our own simulation by making it able to delete the infected packets from our PC, after that we can dump the viruses' packet to prevent them to bypass through our system)). In the ever changing world of global data communications, so for the inexpensive Internet connections and fast-paced software development, security is becoming more and more of an issue. Security is now a basic requirement because global computing is inherently insecure. As your data goes from point to point on the Internet, it may pass through several other points along the way, giving other users the opportunity to intercept and even alter it. It does nothing to protect your data center, other servers in your network or a malicious user with physical access to your system.

Network security: Security is about defense in depth. Providing physical security as well as a well-designed network, control over the users and processes on the host itself and regular maintenance can go a long way towards providing good security. In the most basic sense, a system is secured if it does what it's supposed to do, even if its users attempt to do something they are not supposed to do. It will protects the information stored in it from being modified either maliciously or accidentally by disallowing for reading or modifying by unauthorized users. Security involves tradeoffs. How much is your data worth? Does it make sense to protect your system with the level of security? Since network traffic analysis provides a high level of security in the presence of traffic monitoring. It's a cruel sarcasm in information security that many of the features that make using computers easier or more efficient and the tools used to protect and secure the network can also be used to exploit and compromise the same computers and networks. This is the case with packet sniffing, because once we capture packets using packet sniffer, this packet and its data after that has to be filtered before it goes to be stored in the data base.

Packet sniffer: A packet sniffer, sometimes referred to as a network monitor or network analyzer, can be used legally by a network or system administrator to monitor and troubleshoot network traffic. The packet sniffer simply is able to sniff and captures all of the packets of data that pass through a given network interface. Typically, the packet sniffer would only capture packets that were intended for the machine in question. However, if placed into promiscuous mode, the packet sniffer is also capable of capturing all packets traversing the network regardless of destination. A packet sniffer can only capture packet information within a given subnet. So, it's not possible for a malicious attacker to place a packet sniffer on their home ISP network and capture network traffic from inside your corporate network (although there are ways that exist to more or less "hijack" services running on your internal network to effectively perform packet sniffing from a remote location).

Types of network filter: We have three types of network traffic filters:

- Traffic Filtering: Traffic filtering is a method used to enhance network security by filtering network traffic based on many types of criteria
- Packet Filtering: Packet Filtering is a method of enhancing network security by examining network packets as they pass through routers or a firewall

and determining whether to pass them on or what else to do with them. Packets may be filtered based on their protocol, sending or receiving port, sending or receiving IP address, or the value of some status bits in the packet. There are two types of packet filtering. One is static and the other is dynamic. Dynamic is more flexible and secure as stated below

Static packet filtering: Does not track the state of network packets and does not know whether a packet is the first, a middle packet or the last packet. It does not know if the traffic is associated with a response to a request or is the start of a request.

Dynamic packet filtering: This will track the state of connections to tell if someone is trying to fool the firewall or router. Dynamic filtering is especially important when UDP traffic is allowed to be passed. It can tell if traffic is associated with a response or request. This type of filtering is much more secure than static packet filtering.

Flexible filters: we can filter out all types of packets that have been come from different types of network. And it also called dynamic filtering because of its security. And can filter traffics by:

- Flexible filter: Packet Filter, Email Filter, Web Access Filter
- By MAC address or IP address
- By port numbers
- By protocols
- By packet size, packet value or packet pattern
- Advanced Boolean rules for complex filter formulas
- Supports multi filters simultaneously
- Tracks filter history
- Shares filter settings between projects

So by using this type of filter we can get much more details about the packets and also there is no packet will be unfiltered or the filtering is not completed because the flexible filter has the ability to filter out all the types of network traffics that are traversing in local area network.

Bi-directional communication: For capturing data packets for analysis on a network computing system includes a sending node and a receiving node connected by a bi-directional communication link where the sending node sends a data transmission to the receiving node on the bi-directional communication link. The receiving node receives the data transmission and verifies the data transmission to determine valid data and invalid data and verifies retransmission of the data verified as invalid data as corresponding valid data. This operation called data verification. Which is mean that an apparatus for capturing data packets for analysis on a network computing system includes a bi-directional communication link connecting at least two nodes including a sending node and a receiving node, each of which sending and receiving nodes communicate uses a hardware protocol.

Related works: 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. Simple Network Management Protocol (SNMP) covers a class of tools such as Multi Router Traffic Graphic (MRTG) and Cricket, which collect counter statistics from network infrastructures and visualizes these statistics by means of graphs. The most common use for these tools is graphing the InOctet and OutOctet counters on router interfaces, which will respectively provide counts of the number of bytes passing in and out of the interface. TCPdump prints out the headers of packets on a network interface that match the Boolean expression. It is a network sniffer with in-built filtering capabilities; it can only collect the data from the network, but does not analyze collected data. The collected data can be analyzed offline with another utility namely, TCPshow and TCPtrace. As useful and powerful as TCPdump is, it is only suitable for troubleshooting i.e., for tracking network and protocol related connectivity problems.

MRTG is a versatile tool for graphing network data. This tool can run on a Web server. Every five minutes, it reads the inbound and outbound octet counter of the gateway router and then logs the data to generate graphs for web pages. These graphs can be viewed using a web browser. Although MRTG gives a graphical overview, it however does not give details about the host and protocol responsible for the traffic monitored. Windmill is a modular system for

monitoring network protocol events; it is useful for acquiring the data from the network, but it is however limited in its capability by not providing any facility to aid in the analysis of those events or non protocol events acquired. WebTrafMon uses a probe to extract data from network packets and composes log files. Analysis results are based on the collected log files. Furthermore the user is able to view the analysis result via a generic web browser. WebTrafMon can show traffic information according to the source and destination host through any web interface; it can also show the traffic status according to each protocol in use. Although WebTrafMon has good capabilities, it cannot monitor and analyze traffic in a switched network such as Fast Ethernet and Gigabit Ethernet.

NetFlow Analyzer is a web-based network traffic analyzer using NetFlow devices to generate instant reports on network traffic and users. Cisco Netflow offer the granular and accurate details on applications, users and conversations generating traffic on the network. NetFlow Analyzer collects these Netflow exports, correlates them and generates graphs and reports that help in understanding and troubleshooting network traffic. NetFlow Analyzer allows you to monitor bandwidth and traffic in an interface specific level with one minute granularity. NetFlow analyzer also shows the data points, which gives the traffic IN and traffic OUT details such as speed, volume, packets and utilization off the total bandwidth.

And this way exactly what we have followed and depends for designing our system structure. We used the same concept of Netflow analysis and packet sniffer to capture the traffics and to do the analysis, but the main different is that we made our system completely able to detect suspicious activities such as denial of service or any other viruses by adding into the structure additional function which is Signature Matching, this is the responsible for checking the packet signature after the filtering process to determine whether the traffic has normal or abnormal behavior. And we have tested our system by applying or using packet generator which is the responsible to generate traffic that suppose to have the same virus signature. Also we have made another procedure when the system will capture the infected or the unwanted traffics.

Implementation: We started our study by using smart sniffer that allows us to view the TCP/IP conversation in Ascii mode (for text-based protocols, like HTTP, SMTP, POP3 and FTP). Or as hex dump. (For non-text based protocols, like DNS) and we used WinPcap Capture driver that allows you to capture TCP/IP packets on all windows operating systems. In order to use it, we have to download and install WinPcap capture drive into our PC to make it able to recognize all the packets that are traversing over the network. This method is the generally the preferred way to capture TCP/IP packets with Smart Sniff and it works better than RAW Sockets method. The packet sniffer can capture TCP/IP packets on any 32-bit windows operating system (Windows 98/ME/NT/2000/ XP) as long as WinPcap capture drive is installed and works properly with your network adapter. So, in this study we used the packet sniffer and we completely depended on WinPcap capture driver to identify all the captured packets in our system. And we achieved it by developing a programming code and also we have created an interface to display the results and the graphs that we are going to get from this analysis.

MATERIALS AND METHODS

System design: Our method that we used in this study starting from building an IP structure in our system then we downloaded the packet decode from open source which is the responsible for decoding the data packet and view their headers and payload and so on. And the simulation has been done using visual C++ as simulation code to develop the program structure and to design our system interface as well. And we can get more convincible results by using packet sniffer into our simulation that allows you to capture TCP/IP packets the pass through our network adapter and the captured data as sequence of conversations between client and server. And you can use any operating system to run the execution file, but first we have to download and install the WinPcap capture driver into our PC.

By this software our system will be able to identify all types of network packets which are traversing over the network. And also can identify the protocols that have been used to send them. By using Packet Decode we can get much more information that can help us to analyze the traffics in depth and to understand how it is working by viewing its header any payload using specific parameters in each step in our system structure. From the flowchart below we can see the steps used to design the infrastructure of this system and how it is automated capture and filter traffics.

((The parameters in this system are depends on each other, because from the below Fig. 1 system design we can see that each step is depend on the previous one. So in case of failure in one of the steps, the next step will not be able to analyze the things 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)).

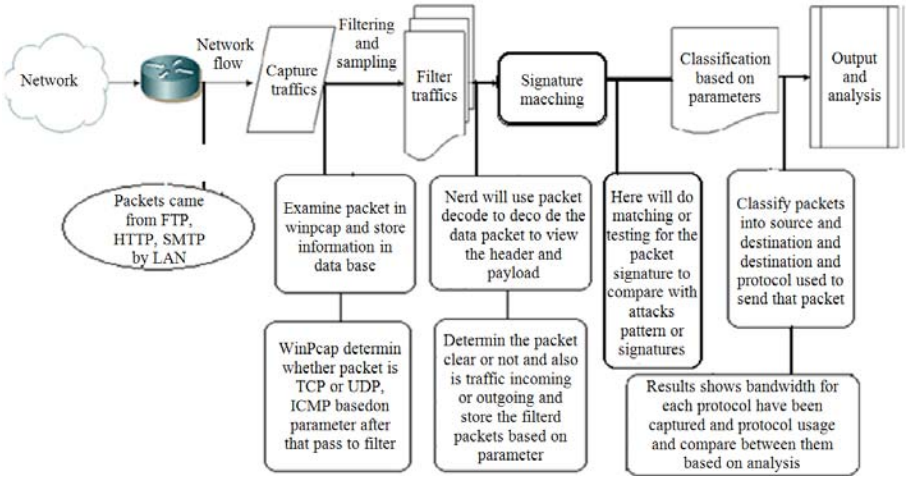

Fig. 1: architecture of the system designs an automated captures and for filter traffics

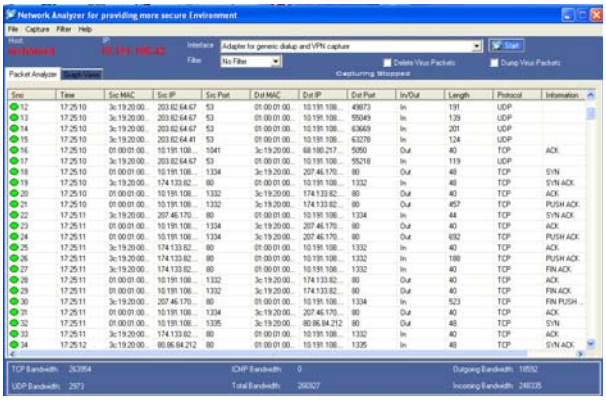

Fig. 2: Main interface displaying the captured traffics

((Here is the explanation of the above system design, So the first step for the system design is)):

- Capture traffics: This will allow you to monitor and capture all the traffics in local area network LAN. So it could be in university or star-bucks or home or in anywhere that you can find internet connection. in this step there is a very important process which is related to WinPcap, the WinPcap in this step will examine the packets with its stored data base, so the packet data are suppose to be matched with the stored WinPcap data base to determine the packet type and the protocol used to send it, it could be (TCP or UDP, ICMP, IGMP). So once it finish, it will forward the information to the subsequent step
- Filter traffics: After finishing capturing process it will forward the packet to the next step be filtered. In this step the system will use here the packets decode to reveal the packet headers and payload and will compare the decoding results with the information sent from the WinPcap to determine whether the traffic is normal or abnormal. After that it will arrange the information got from the analysis such as the packet source and destination and send it to the next step for determining the packet activities and behavior
- Signature matching: In this step after the packet becomes filtered it will give in to check the packet signature. Here is the most critical point, because in this step the packet will be determined whether its signature is normal traffic signature, or its attacks signature
- Classification based on parameters: Here the system has finished analysis and the traffics have to be ready to view in details such as the packet:
- Source and destination and the protocol used to send the frame as so on.
- Results and Analysis: The results shows the bandwidth rates for each protocol based on its usage in the network and view the results in graph to make a reasonable comparison between them.

So, we have illustrated these operations into the system interface that we have designed to view all traffics information and details such as traffic source and destination, IP source and destination and the protocol used to send them and the system will display all the data that have been gathered and view it as results and you can design a display filter to specify how much information that you have captured will be displayed in Network Monitor's Frame Viewer window as it is shown in this Fig. 2.

By using packet decode we can get all this information and other additional information such as bandwidth and TTL and the checksum for both of the IP layer and the protocol layer (TCP. UDP, ICMP, IGMP) and also we can view the data packet in hex. The process by which Network analyzer collects this information is called capturing.

RESULTS

The results that we are going to show is actually what we have got from decoding the data packets then from the packet generator, we will show how we generated the traffic the has same intruders or hackers signature to be detected by our system as abnormal traffic. And also the results for the protocols graph comparison are shown after the software will finalized the above steps.

Protocol packet decodes: After the network packet has been captured, it will forward to be filtered, in the filter operation we will use packet decode to analyze the packet in depth. The information that we are going to get from decoding the packet as follow:

- Decodes packet headers for the often used TCP/IP protocols and applications
- Shows packet data in HEX, ASCII and EBCDIC
- Reveals message protocols and data sizes
- Displays IP addresses and MAC addresses of captured hosts
- Reconstructs TCP streams into original messages

Most of the information that we have got from this analysis has an accurate details for TCP more than UDP when the TCP structure is very complex comparing to that one in UDP, so here we will show some reminder and comparison between the TCP and UDP to understand what we are going to face when will decode the packets. (TCP and UDP are both protocols that run on top of IP. TCP has guaranteed delivery and UDP does not. You would select one or the other for port forwarding depending on what service you're trying to forward. IITTP, for instance is TCP. If you don't know what protocol the service you're trying to forward is, it's almost certainly TCP). From Fig. 2, we can see that the system is started to capture packets and view the initial information such as source, destination and protocol and so on. By using packet decode as we mentioned earlier we can get all this information. And by design a specific level of simulation we can view the header for each packet in hex and decimal and much more information such as time to leave for each packet and checksum and other information, if we want to decode any of the captured packets to view its details in hex and structure as well, we have to stop the capture process and select the packet that we want to decode after that from the file selection on the top of the interface then decode packet. The results that we suppose to get will be shown in the next Fig. 3.

As we can see from the above illustration 5.1, we have chose packet number 1 to be decoded and as it is shown from the small interface which is inside the main interface, this interface has the responsibility to view the decoded packet in packet structure type and packet in hexadecimal type. All these functions and more other functions are found to help users to understand and comprehend the purpose of Network Traffic Analysis and why we do need to use it. So, now we have all the details and information about the packets that we have captured and examined to determine whether this packet has normal or abnormal data packet. After that we can safely download them to keep it for future need.

Packet generator: Now we will discuss about how we can detect an intruder or a suspicious activity by our system because as we have mentioned before that this rarely when happened. Now to detect or to apply this function we will use the traffic generator or packet generator, by download or purchase it from the internet. And this packet suppose to contain a specific or special specifications (signatures) chosen by users such as (viruses, unwanted program, abnormal activity, or infected program) and we will push it and make it pass through our system adapter which supposed to be able to detect and prevent our network environment from like this kind of traffics or threats. Viruses and hacker attacks typically generate a recognizable pattern or “signature” of packets. A network analyzer can identify these packets and alert the administrator to their presence on the network via email or page.

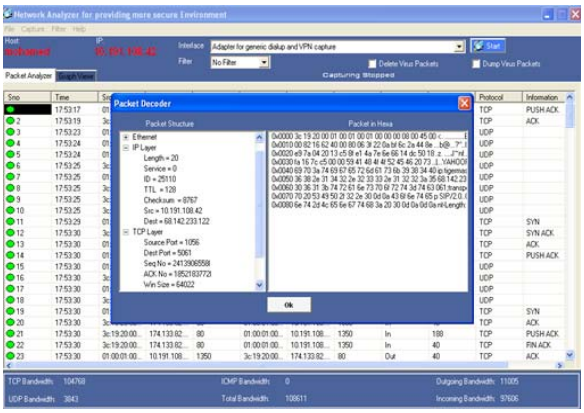

Fig. 3: Displaying for the results from the packet decoder

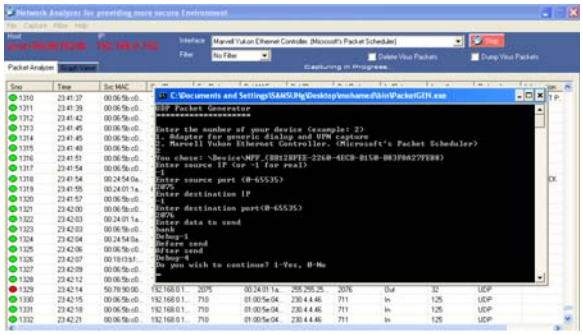

Fig. 4: Generating a network traffic using command prompt and this traffic suppose to be appeared as an abnormal traffic

We want to illustrate this situation by applying packet generator that will be acting the viruses or hackers activities. In order to send the traffic or the packet that we are going to generate, this is completely depending on WinPcap library which has the responsibility to recognize all the types of packets. So to generate a packet and to make our system able to identify it we have first to coordinate with the WinPcap to allow us to generate that packet (this is the WinPcap allowance or permission issues). For more explanations, we have to know how WinPcap is able to identify all types of packets. So, because of the majority of the packets or traffics that are traversing over the network has to be registered in WinPcap.

So, the WinPcap should have all the packet details and events, such as protocols and original size and the registration time for these packets. Once the WinPcap will recognize the packet, it will send the information to the stored database to check whether this packet still valid or not. If not, the WinPcap will notify us by the existence of illegal packet. If the packet still valid it will forward it to the next step to be decoded and make another check for the data or the packet content. So from these explanations, we can imagine how the system has a high level of accuracy while checking the packet. Because the packet will be examined and tested in two stages, one in the WinPcap itself to determine the validation of the packet and another one with Packet decode which is located within the packet filter to determine whether the content of the packet is clear or not. By developing a specific simulation to generate the packet or the traffic we have low level of permission from the WinPcap to over control the packet that we are going to generate and we are going to use the command prompt to generate the traffic by writing a specific information we have prepared earlier that being coordinated with WinPcap. So once we generate the traffic using the command prompt the WinPcap will check the traffic and the packet decode will check the information inside it and it have to be matched as a virus signature to identify it as abnormal packet and the next illustration 5.2 will show the packet generating operation using command prompt.

Now, as we can see from Fig. 4, it's illustrating how to generate traffics using command prompt and the type of information that have to be inserted in it, such as Enter the number of the device (number of interface) and source IP, destination IP, source port, destination port and the data that we want to send. All these information given have to be matched with the information that we have prepared earlier. Now we have already generated the traffic and our traffic has been successfully appeared as abnormal traffic by the red color indication as it is shown in Fig. 4. Now we can easily delete that packet by click on the delete virus packets button, it will delete all the viruses' packets from the list and from our PC as well. Now network traffic analysis can be study as an antivirus in additional to its main functions to capture the traffics and do the analysis and generate the graph and so on to keep Our PC protected from any intruder during the capturing or analysis. In order to modify the infected data packet to make it valid or to drop it from the network to ensure that this packet will not affect our network environment anymore we have to purchase an original version of WinPcap which allow us to modify or to drop the infected packets.

Deleting abnormal traffic: Now we have already generated the traffic and our traffic has successfully appeared as abnormal traffic by the red color indication in Fig. 2. Now we can easily delete that packet by click on the delete virus packets button, it will delete all the viruses' packets from the list and from our PC as well. By using packet generator we are able to generate more than one packet by filling traffic information such as another source IP, destination IP and source port, destination port and data. So we will have two packets have been generated and have the same information that we gave and both of them have appeared abnormal as it is shown in this Fig. 5.

Once we have generated the traffics and they successfully appeared as abnormal traffics we can easily delete them from our PC and from the list as well by click on the delete virus packets button it has to be deleted after that.

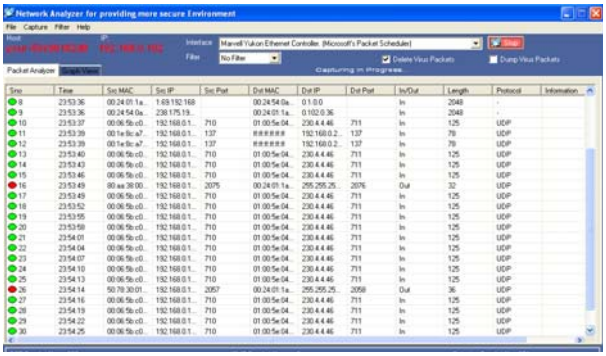

Fig. 5: Generating more than one traffic using packet generator

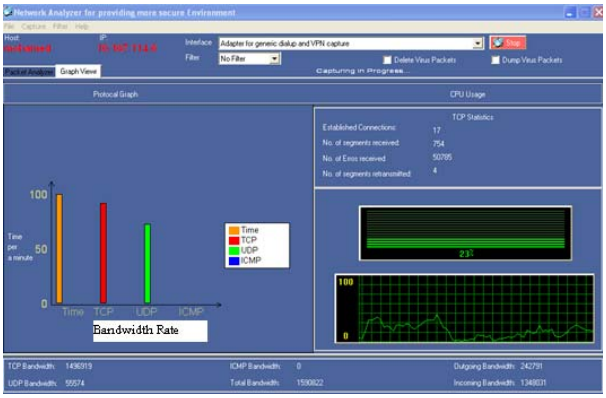

Fig. 6: The second interface displaying the protocols graph and CPU usage

Protocols graph (TCP and UDP): Now we will start with the second interface to view the graph which is responsible to examine the bandwidth for TCP and UDP and ICMP per minute. In order to get reasonable comparison between the protocols that we have mentioned above we have to examine packets or let the capture in progress for a long time off period to overcome the high rate of the transient packets. And also the graph generator actually is working based on specific parameters for doing simple calculation for the bandwidth per minute, so we will observe a different bandwidth rate in the graph from minute to another. To justify this different, as we mentioned above the graph generator is recalculate the bandwidth every minute, so when it will do the calculations for the subsequent minute it has to start from zero and ignoring the calculation for the previous minute.

In other word, when we do the browsing from website to another while the system is running and the capture is in progress, the calculation will be based on that site, that particular site could has the UDP users rate higher than the TCP users rate and the contrary is true. That is why we may observe the UDP bandwidth rate sometime is higher than TCP. So, once we run the software after a certain period of time we can get the approximate rate of bandwidth as it is shown in Fig. 6 from the graph viewer interface.

As we can see from the protocol graph above, it's examining the bandwidth for the protocols that have been listed in the graph. And we observed that the different between TCP and UDP is not that much in a normal case or by randomly browsing means by jumping from one site to another, while ICMP is seldom finding. Also we can see from the CPU Usage part, we have found the above calculation is ready in the open source to calculate out the established connections and the number of segments received and other information for TCP Statistics only, because of TCP is a transmission control protocol, Means that its oriented connection protocol, while other protocols such as UDP or ICMP or IGMP is connectionless protocol. And as it is shown in the next graphs we have another testing for the bandwidth rate after a long period of time. The graph 5.5 and 5.6 involved to show the new results in different period of time and each graph will express about specific site that we browsed at the same time when the system has done the calculation and each site has its protocols users those are the most suitable or more preferred to be used in this site because of the compatibility between that particular site and the massages or the frames that the site's users sending usually. So, the examples are shown below.

In the graph 5.5, we can see the comparison between UDP and TCP for bandwidth rate and how the UDP now is higher than TCP, because its completely depending on the proportion of the protocol that have been used in the site that we are browsing at the same time when the system has done the calculation. So, once we do any browsing in the internet, the protocols graph reading will be different from one minute to another and the graph will be plotted according to the site that we are browsing.

For example, Google search engine has more UDP usage than other protocols as it is shown in Fig. 7. And Fig. 8 shows the bandwidth results comparison between TCP and UDP protocol in Yahoo search engine and we can see that yahoo search engine has both of TCP and UDP protocol users and usually TCP is more than UDP in that site. So by the results shown in the system graph, we can use it to determine or to know which sites has more protocols users and what is the protocol used more than other in that particular site.

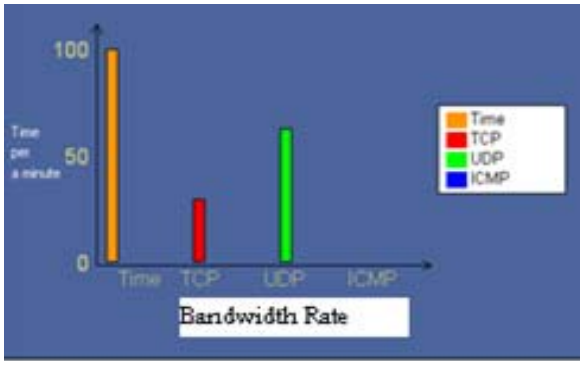

Fig. 7: Bandwidth results comparison between TCP and UDP protocol while browsing different sites

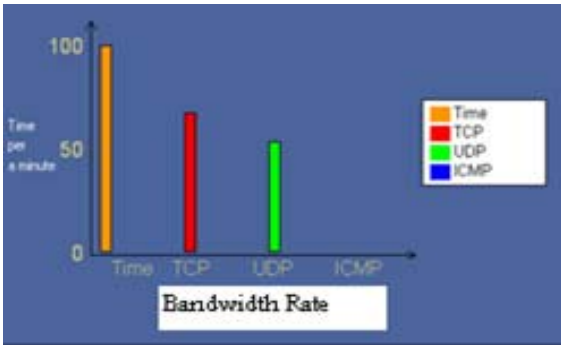

Fig. 8: Bandwidth results comparison between TCP and UDP protocol while browsing different sites

CONCLUSION

Network Analyzers will never replace your firewall, anti-virus software or intrusion detection system or intrusion prevention system. However, because it is not possible for these precautions to be completely effective, you cannot maintain the security of your network without a network analyzer. A good analyzer alerts you when the other defenses have failed and takes much of the pain out of identifying, isolating and cleaning up compromised machines. According to the results shows in the graphs, we observed that the different between TCP and UDP is not that much in a general case, while ICMP is seldom finding. And the difference rate between the TCP and UDP is depending on the website. If the website has a UDP users rate more than TCP, or in the contrary, the graph will show us after a few seconds the increasing rate in the protocol used in that website.

We have concluded also that by using packet generator that will generating a traffics that suppose to has the same signatures for the intruders or hackers, we found that our system is able to detect any kind of threats or suspicious activities and the deletion operation for the infected packets has been done successfully. For future study we want to make the system able to capture traffics from both the lPv4 and lPv6, especially detecting suspicious activities. Most of the packets that are traversing over the network are using IPv4. And for the hug establishments or in the industry now days they are using lPv6 in average of 10%, because it has bigger address space than lPv4. For more advanced level we can use a specific simulation code to allow our system to be updated monthly for the new abnormal traffics signatures.

REFERENCES

Aggarwal, C.C., J. Han, J. Wang and P.S. Yu, 2003. A framework for clustering evolving data streams. Proceedings of the 29th International Conference on Very Large Data Bases (VLDB'03), VLDB Endowment, Germany, pp: 81-92. http://portal.acm.org/citation.cfm?id=1315460

Almulhem, A. and I. Traore, 2005a. Experience with engineering a network forensics system. Proceedings of the International Conference on Information Networking (ICOIN 2005), Korea, LNCS 3391, Springer-Verlag, Berlin, Heidelberg, pp: 62-71.

Almulhem, A. and I. Traore, 2005b. Experience with engineering a network forensics system. Lecture Notes Comput. Sci., 3391: 62-71. DOI: 10.1007/978-3-540-30582-8_7

Almulhem, A., 2009. Network forensics: Notions and challenges. Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Dec. 14-17, IEEE Xplore Press, Ajman., pp: 463-466. DOI: 10.1109/ISSPIT.2009.5407485

Anaya, E.A., M. Nakano-Miyatake and H.M.P. Meana, 2009. Network forensics with neurofuzzy techniques. Proceedings of the 52nd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2009), Aug. 2-5, IEEE Xplore Press, Cancun, pp: 848-852. DOI: 10.1109/MWSCAS.2009.5235900

Ashfaq, A.B. M.J. Robert, A. Mumtaz, M.Q. Ali and A. Sajjad *et al*., 2008. A comparative evaluation of anomaly detectors under portscan attacks. Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection, Springer-Verlag, Berlin, Heidelberg, pp: 351-371. DOI: 10.1007/978-3-540-87403-4_19

Babcock, B., M. Datar, R. Motvani, 2003. Load Shedding Techniques for Data Stream Systems. Stanford University. http://www.inforsec.org.cn/dataflowgroup/dsms_home_en/Reading/2003paper/babcock03a.pdf

Babcock, B., M. Datar, R. Motwani and L. O'Callaghan, 2003. Maintaining Variance and k-Medians over Data Stream Windows. Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM Press, New York, USA., pp: 234-243. DOI: 10.1145/773153.773176

Bon, K.S., 2009. Integrating intrusion alert information to aid forensic explanation: An analytical intrusion detection framework for distributive IDS. Inform. Fusion, 10: 325-341. DOI: 10.1016/J.INFFUS.2009.01.001

Chen, T. M., 2004. Intrusion detection for viruses and worms. Southern Methodist University, http://www3.engr.smu.edu/~tchen/papers/iec2004.pdf

Datar, M., A. Gionis, P. Indyk and R. Motwani, 2002. Maintaining stream statistics over sliding windows. SIAM J. Comput., 31: 1794-1813. DOI: 10.1137/S0097539701398363

Domingos, P. and G. Hulten, 2000. Mining high-speed data streams. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, USA., pp: 71-80. DOI: 10.1145/347090.347107

Ganti, V., J. Gehrke and R. Ramakrishnan, 2002. Mining data streams under block evolution. SIGKDD Explorat. News Lett., 3: 1-10. DOI: 10.1145/507515.507517

O'Callaghan, L., N. Mishra, A. Meyerson, S. Guha and R. Motwani, 2002. Streaming-data algorithms for high-quality clustering. Proceedings of 18th International Conference on Data Engineering, Feb. 26-1 Mar., IEEE Xplore Press, San Jose, CA , USA., 685-694. DOI: 10.1109/ICDE.2002.994785

Riech, K. and P. Laskov, 2006. Language models for detection of unknown attacks in network traffic. J. Comput. Virol., 2: 243-256. DOI: 10.1007/s11416-006-0030-0

SANS, 2007. Top 20 Internet Security Problems, Threats and Risks. SANS Institute. http://www.sans.org/top20/2007/

Tan, L. and T. Sherwood, 2006. Architectures for bit-split string scanning in intrusion detection. IEEE Micro., 26: 110-117. DOI: 10.1109/MM.2006.5

Wang, W. and E.D. Thomas, 2008. A graph based approach toward network forensics analysis. ACM Trans. Inform. Syst. Sec., 12. DOI: 10.1145/1410234.1410238