The International Journal Of Engineering And Science (IJES)
|| Volume || 3 || Issue || 10 || Pages || 01-09 || 2014 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Implementing a Robust Network-Based Intrusion Detection System

[1]Ogheneovo, E. E. ,[2]Japheth, B. R.
[1]*Senior Lecturer, Department of Computer Science, University of Port Harcourt, Port Harcourt, Nigeria.*
[2]*Lecturer, Department of Maths/Computer Science, Niger-Delta University, Yenagoa, Nigeria.*

----------**ABSTRACT**----------
Information security is of great concern these days due to the activities of hackers and malicious users on the Internet. Securing information has become a critical issue and is of growing concern as computer systems worldwide become increasingly vulnerable to the rapid increase in the volume of information being transmitted across networks and over the Internet. In this paper, we proposed a technique that provides a robust intrusion detection system against attackers. It puts scalability and appropriate security in mind. The framework is made up of a Network Intrusion Detection System (NIDS) for detection of traffic to and from a given network or sub network, a Host based Intrusion Detection System (HIDS) and a line for possible Intrusion Prevention Systems (IPS). The technique is implemented by modeling network using OPNET, a network simulation software; since a real life implementation is very costly. Our result shows that the technique provides an intrusion detection system that can be used to monitor user's activities on the Internet and other networks with relatively minimal false alarms.

KEYWORDS: Intrusion Detection Systems, intrusion detection, HIDS, NIDS, network, hackers.

Date of Submission: 28 July 2014 Date of Publication: 30 October 2014

I. INTRODUCTION

The Internet no doubt has revolutionized the world in recent times. As a result, businesses have become more open to individuals who want to buy or shop for goods and services. These services include customer care, e-commerce, and extranet collaboration, sourcing for information, etc. Due to the advantages the Internet offers, many people have been using it for bad motives such as gaining access to people's Web sites and accessing information without authorization. As a result, the Internet and other enterprise networks have been broken into by hackers. For instance, the US Citibank reported a security breach in 1994. This caused about 10million dollars loss in revenue. Only 400,000 dollars was eventually recovered [1]. Joseph [2] also observed that there were a high number of unauthorized security events, and in the year 2000 alone, 70 percent of organizations in the US at least reported security breach to their computers. This represents 42 percent increase from the 1996 report. According to them, Computer Emergency Response Team (CERT) reported 3734 incidents in 1998, 9859 in 1999 and within only the first six months of 2000, 8836 incidences where already reported. These are but just a few of the identified and published forms of computer system attacks. As a result, there is need for an intrusion detection system that can be used to monitor user's activities on the Internet and other networks.

Intrusion detection [3] [4] is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or security standard practices [5] [6]. To understand the meaning of intrusion detection, we can use an analogy to the common "burglar alarm". Just like the burglar alarm, intrusion detection works on a computer system or network and is enabled to detect possible violations of security policies and raise an alarm to notify the proper authority. Intrusion Detection Systems (IDS) looks for signatures, which are specific patterns that usually indicate suspicious or specific patterns that usually indicate suspicious or malicious intent [7]. IDS is used to detect malicious activities that compose the security of a computer system.

An attacker can craft a legitimate HTTP request in such a way that it will look legitimate and perform the evil actions. The widespread use of information stored and processed on network-based systems in most businesses has increased the necessity of protecting these systems [8]. Most businesses are constantly experiencing new threats and vulnerabilities in their applications. Therefore, trying to keep up with emerging

threats, applying patches against known vulnerabilities, updating antivirus software, updating firewall rules and all of the other security measures can have a network or security administrator working 24 hour days, 7 days a week with no vacation. There is a crucial need to address security issues that affect networks. It is equally important to be able to sift through the mountains of potential threats and determine which ones truly affect your network so that your time and resources can be put to the most efficient use.

Organizations are striving to maintain confidentiality, integrity and availability of their networked resources and a number of techniques have been employed to guard against network intrusion. However, even though these measures provide some level of security, they have been found to be lacking in a number of ways. In the past, firewalls have been used but they have been found not to provide enough security as it is easily compromised. It is mostly used to control traffic to or from a private network. Firewall, user authentication, data encryption and Virtual Private Networks (VPN) provide a level of security but they are limited by the fact that they cannot give protection against malicious codes, inside attacks or unsecured modems [9] [10]. They therefore would only be effective as one of the available lines of defense. For institutions that already have intrusion prevention systems, perfectly secure system are hard to come by. There are always a number of system flaws in addition to possible administrator configuration errors. Intrusion detection systems can thus be used to supplement the already existing systems.

In this paper, we design an effective and scalable intrusion detection system that can be used in an organization such as an academic institution with emphasis on network security. Although, we implemented this work by using modeling technique due to the costs of implementing it in a real life situation, however, we are sure if applied to real life situation will also be very effective and functional. The rest of the paper is as follows. Section 2 discussed related work; section 3 discussed the methodology adopted, the architecture of the system and the simulation of the network-based IDS. Section 4 discussed the simulation result and also the discussion of the result, and section 5 draws a conclusion.

II. RELATED WORK

Daniels et al. [11] proposed the distributed intrusion detection systems as a system that uses varying techniques to combine elements of both Network-based Intrusion Detection System (NIDS) and Host-Based Intrusion Detection System (HIDS).They are designed to maximize on the strengths of NIDS and HIDS, while minimizing on their weaknesses. This could possibly happen in high security networks such as server farms and they may also be implemented in settings where some hosts on the same network will require relatively more security and therefore in addition to NIDS, HIDS may be installed in them. Ibrahim et al. [12] proposed a technique that uses two different approaches to intrusion detection system, phase and level approaches. The phase approach comprises three phases. Phase 1 accepts the input data and check if it is an attack, phase 2 classifies the attack while phase 3 records the attack into the appropriate classification type. The level approach also has 3 stags. Level 1 detects normal and attack profiles, level 2 records and classifies the attacks into 4 categories; level 3 classifies each attack type and records them. However, the model did not address the problem of countering intrusion attacks. In situations where there are intensive attacks, not only will actual alerts be mixed with false alerts, but the amount of alerts will also become unmanageable.

Ning et al. [13] proposed a technique for constructing attack scenarios through alert correlation using prerequisites and consequences of attack. This approach is based on the fact that a series of attacks, that were not isolated but are based on earlier stages as a prerequisite for the next stages. The framework intuitive representation is based on correlated alerts that reveal the attack scenario of the corresponding attacks. A set of interactive utilities was also developed to facilitate the analysis of large sets of correlated alerts, including hyper-alert generating utilities to reduce the scope of alerts being investigated, and feature extraction utilities to facilitate the examination of some properties of some selected sets of alerts. Faizal [14] proposed a technique for verifying threshold value for network intrusion detection system especially in detecting fast attacks. The threshold value is obtained using observation and experimental techniques. The results from both techniques are then compared and verified using statistical control process approach [15] [16] [17]. Real time network data, simulated data from DARPA99 and data obtained from the experimental setup, any connection that exceeds the threshold value of 3 within 1 second is considered abnormal and harmful.

III. METHODOLOGY

The design takes into account the knowledge of how dynamic the Information Technology (I.T) industry has been. It puts scalability and appropriate security in mind. The framework is made up of a Network Intrusion Detection System (NIDS) for detection of traffic to and from a given network or sub network, a Host based Intrusion Detection System (HIDS) and a line for possible Intrusion Prevention Systems (IPS). The NIDS

detects against network level intrusion and since traffic such as encrypted data may not be detected by the NIDS, host level Intrusion detection systems to further enhance intrusion detection. The framework provides a flexible approach that will enable the network designer put in place prevention mechanism that will enhance intrusion detection and meet the required level of security while at the same time trying to strike a balance between security and resource utilization. Each of these has rules to refer to in determining which traffic is unwanted.

The intrusion detection system is configured behind a firewall. This architecture helps detect any of the traffic that may have bypassed the firewall. It also has a Dynamic Host Configuration Protocol (DHCP) machine. It also decreases on the workload for the IDS and hence IDS efficiency. The DHCP if not well protected by a good line defense or intrusion prevention techniques can be compromised by the attacker through denial of service attacks. The DHCP also by itself may not force clients to release their leases at shutdown. Malicious hosts for example may therefore continue to allocate new addresses without releasing them at all, leading to exhaustion of addresses. Also, when external clients are intentionally or accidentally configured to use addresses of internal clients, regulation of addresses will become hard and therefore leads to a security disaster. Figure 1 shows an architectural framework of the proposed intrusion detection system.

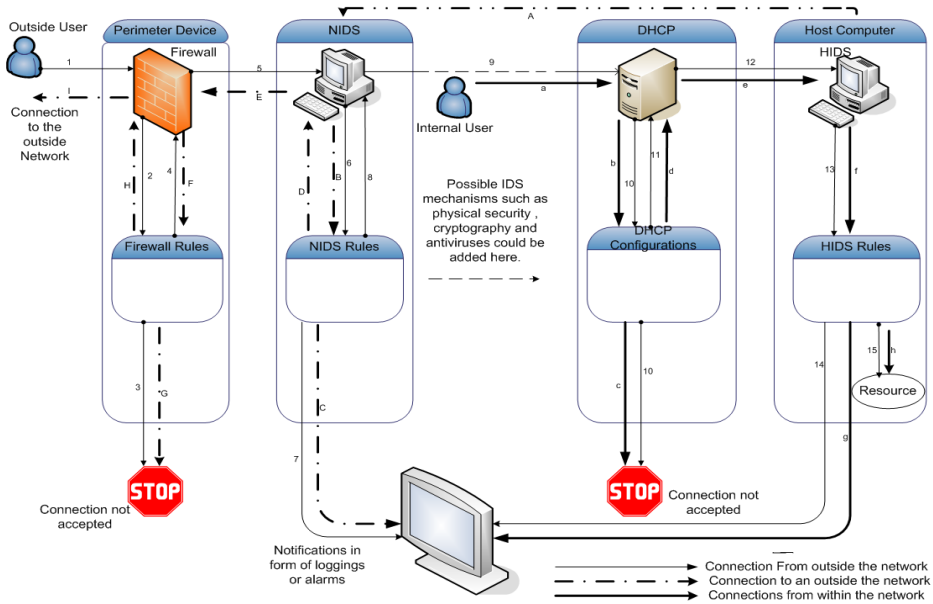

Fig. 1: Architectural Framework of the intrusion detection system

The design of this architecture takes into account the knowledge of how dynamic the Information Technology (I.T) industry has been. It considers the concepts of scalability and appropriate security for the network. The framework is made up of a Network Intrusion Detection System (NIDS) and a Host-based Intrusion Detection System (HIDS). The NIDS is used for detecting traffic to and from a given network or sub network while the (HIDS) is used for Intrusion Prevention Systems (IPS). The NIDS detects against network level intrusion and since traffic such as encrypted data may not be detected by the NIDS, host-based Intrusion detection system is used to further enhance intrusion detection by providing a two-level security. The framework provides a flexible approach that will enable the network designer put in place prevention mechanism that will enhance intrusion detection and meet the required level of security while at the same time trying to strike a balance between security and resource utilization. Each of these has rules to refer to in determining which traffic is unwanted.

3.1 Simulation of Network-Based Intrusion Detection Systems

In this analysis, we are assuming that there is an attack to a network and the attack is simulated in OPNET. Figure 2 shows the OPNET model for simulating the attack. There are 10 virtual PC nodes arranged into two columns in the figure: PC 0 – 4 on the left side and PC 5 – 9 on the right side. The top node in the center column is the "generator", which prepares the packets extracted from the traffic source. Once a packet is ready, it is given to its source PC node, and from there it will be sent to the destination PC node through the hub (located at the bottom of the center column). There is no delay between the generator and the end PC nodes, so

the traffic flow is consistent with the captured traffic source. The number of the virtual PCs is the outcome of preprocessing the source traffic file. Since there are 10 distinct IP addresses in the source, the model uses 10 PC nodes connected to each other through a hub. Node 0 (the top node in the left side column) is the "hacker", and node 1 (below node 0 in the figure) is the "victim" of the attack. There is a "firewall" node between the victim and the hub which we use to capture suspicious data packets to or from the victim using the attack's signature.

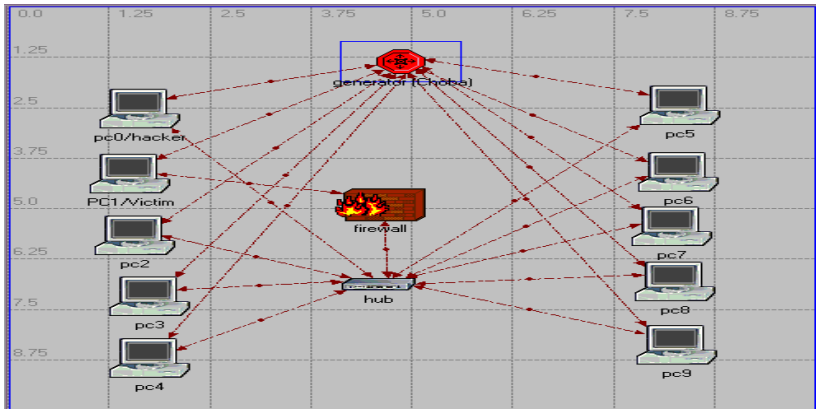

Fig. 2: The network model simulating Uniport intrusion

The node domain of the "generator" is shown in Figure 3. There is a generator module (pk_generator) configured to use a script file of inter-arrival times for generating the packets. The script file is the result of pre-processing the traffic source. Figure 4 shows the attribute panel of pk_generator.

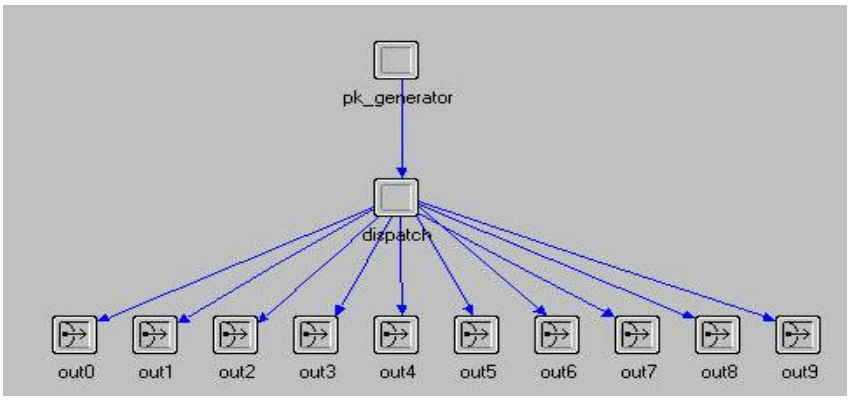

Fig 3: The node structure of the packet generator

Attribute	Value
name	pk_generator
process model	simple_source_1
icon name	processor
Packet Format	_virtual_pk
Packet Interarrival Time	scripted (ourTimeStampSequence)
Packet Size	constant (1024)
Start Time	0.0
Stop Time	Infinity

Fig. 4: The attribute panel of "pk_generator"

Within the "dispatch" module of the generator node in Figure 4, the traffic source file is parsed and the next data packet extracted. Whenever a packet arrives from "pk_generator", its fields are set according to the corresponding values of the data packet from the source traffic, e.g., the destination, flags, etc. Then, the packet is sent to the PC node corresponding to the source IP address. Thus, the packet arrival time and its contents will match the information as in the original traffic source.

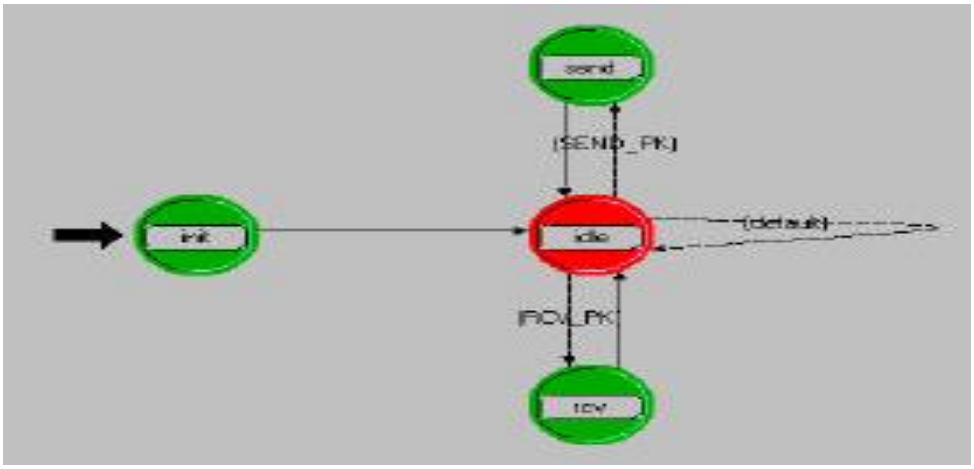

Fig. 5: The structure of the virtual PC in process domain

Figure 5 depicts the process domain for each virtual PC, which supports the packet streams in and out. We also set up a firewall between the hub and the victim of the network attack. The firewall uses a simple signature-based detection which looks for packets sent to port 139 (NetBIOS) of the victim PC with the "urg" flag set in the packet header. The pre-processing tools we developed can be reused for simulating other types of intrusion attacks. To demonstrate, we also simulated the ProcessTable DOS attack using the Uniport ICT Center Lab TCPDUMP files. We needed to set up a network in OPNET using 20 PC nodes because there are 20 distinct IP addresses involved in the traffic source. We also modified the intrusion detection logic of the firewall node using the new attack's signature, and added the corresponding statistical measures to the OPNET simulation. The results of the simulations are described in the following section.

IV. RESULTS

The source traffic data for the network attack comes from the Uniport ICT Center Lab TCPDUMP outside file, 2014/week14/Monday data set. This data set includes the initial 5 minutes of data, and only one type of the attack. In our experiment, we pre-processed the source file, and extracted less than 3 minutes of data containing a total of 367 TCP packets. There were 10 packets captured by the firewall node due to the network attack, 9 of which were sent from the attacker node to the victim and one sent from the victim back to the attacker. We set up several statistical measures in OPNET to study the performance of the intrusion simulation. For example, Figure 7 depicts the IP address distributions of the data packets during the entire simulation, where the IP addresses correspond to the PC node numbers 0 – 9 of the y-axis. The figure clearly demonstrates patterns of consecutive accesses to the same IP addresses during several short intervals of the simulation although these accesses are irrelevant to the Uniport attack.

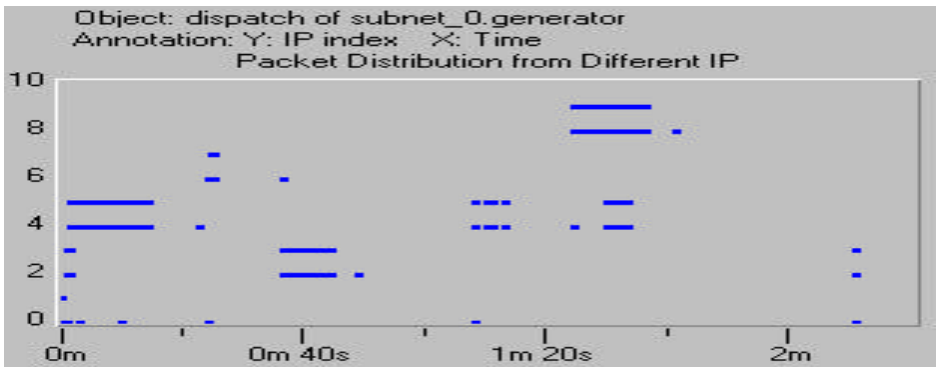

Fig. 6: IP address Distribution of data packets

Figure 6 depicts the rates of data packets captured by the firewall. The occurrences of the packets and the times of their arrivals are clearly shown in the figure – there were a few rapid arrivals in the beginning, followed by 3 more at later times. This figure demonstrates the occurrences of the Uniport attack and its capture by the firewall. This is indicated in figure 7.

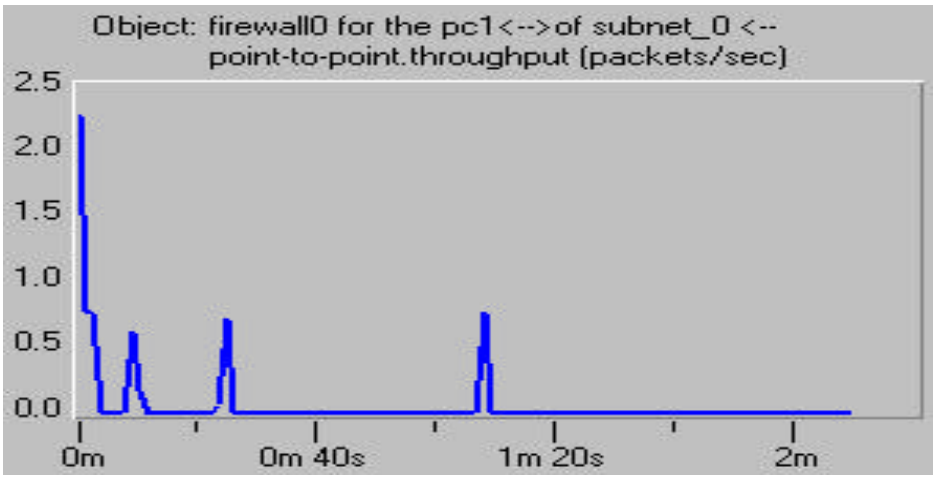

Fig. 7: The inbound traffic of the firewall

We also collected statistics of the overall network traffic, which is depicted in Figure 8, although this performance measure seems irrelevant to the Uniport attack.

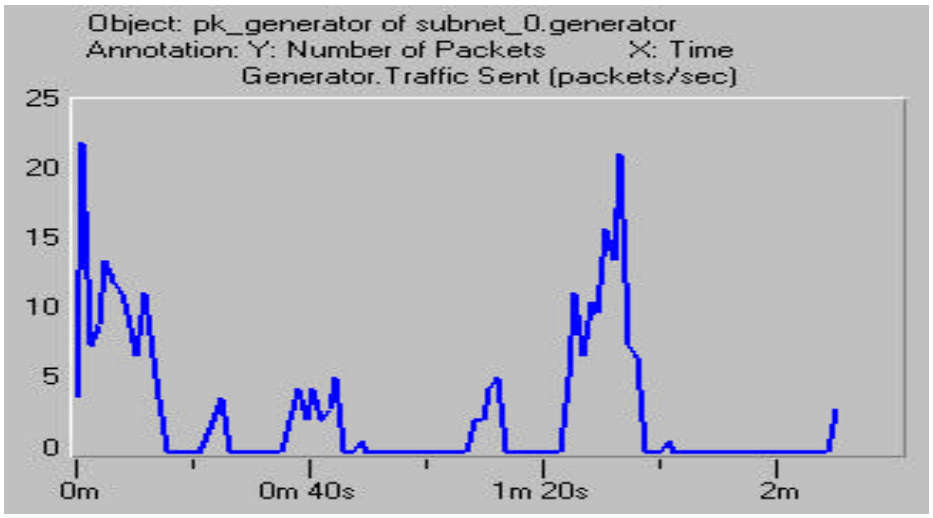

Fig. 8: The overall network traffic during simulation

4.1 The ProcessTable Attack

To demonstrate the reusability of our pre-processing tools and to demonstrate our intrusion simulation methodology, we also simulated the ProcessTable DOS attack. This attack aims at filling up the process table of the underlying operating system, rendering the system lifeless until the attack terminates or when the system administrator kills the attacking processes. The ProcessTable attack can be detected by recording a large number of connections to a particular port of the victim node during a short period of time. In our simulation, we used the Uniport ICT Center Lab TCPDUMP file that contains the ProcessTable attack packets, extracted the pertinent information using our pre-processing tools, and then set up the simulation in OPNET. There are slightly less than 2 minutes of data with a total of 5526 data packets. We collected two statistical measures at the firewall node attempting to detect and identify the ProcessTable attack. Figure 9 depicts the number of distinct port connections to the victim PC during simulation. It can be seen very clearly that there are 3 jumps in the graph, indicting rapid increases of port connections to the victim during 3 distinct time intervals.

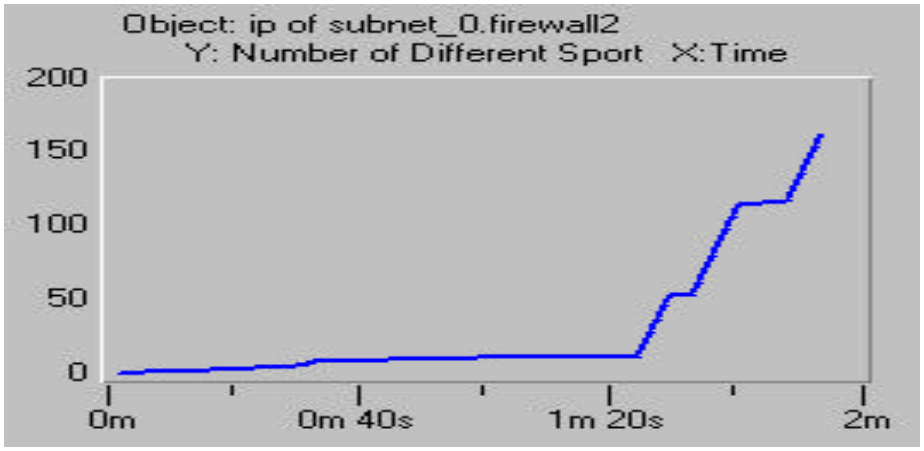

Fig. 4.8.1: Number of distinct port connections to victim

The ProcessTable attack can also be directed at a particular port of the victim. Figure 10 depicts the network traffic directed to Port 25 of the victim during simulation. The graph displays two peaks: the first occurred around the one-minute mark; the second started after one minute 20 seconds and lasted to the end. Thus, the two figures 9 and 10 clearly demonstrated data packets that are suspicious of the ProcessTable (or similar) attacks.

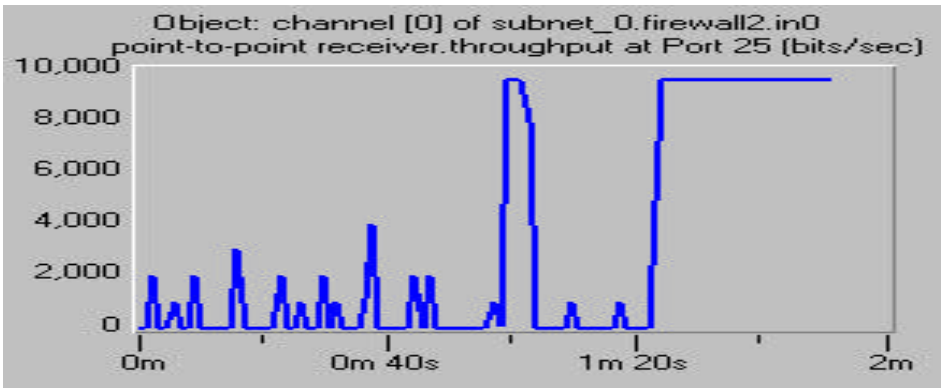

Fig. 10: Data traffic to Port 25 of the victim PC

4.2 Discussion of Results

Another goal of our research of intrusion simulation is to study the simulation efficiency, that is, how to speed up the simulation and intrusion detection of intrusion traffic for large data files. We first used the data file of the Uniport attack and simulated the data packets and intrusion detection of different time durations. All simulations were performed on a Dual Core PC, with a 2.83 GHz CPU and 1024 MB RAM. Figure 11 plots the OPNET simulation time running the data files of durations ranging from 30 seconds through 131 seconds, at an increment of 30 seconds. Since there are only a few hundred data packets (367 exactly), all simulation runs completed within one second.

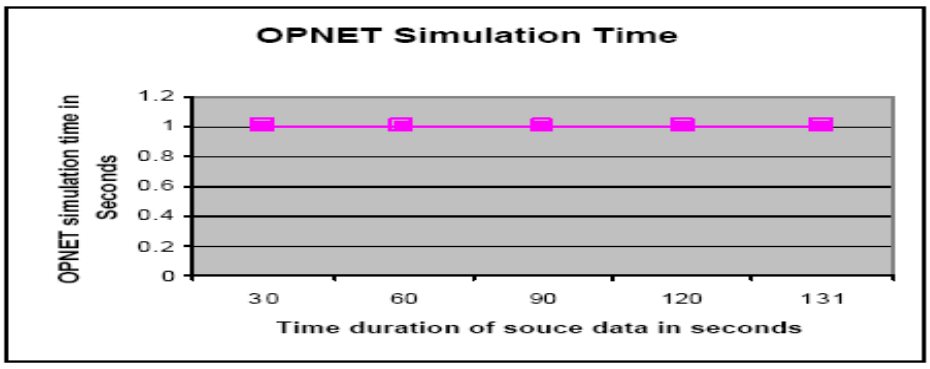

Fig. 11: OPNET simulation time of the Dosnuke attack

We also ran the simulations of the ProcessTable attack file of different time durations to measure the simulation efficiency. Figure 12 plots the OPNET simulation time running the data files of durations ranging from 30 seconds through 114 seconds, at an increment of 30 seconds. There are a total of 5526 data packets in the entire file (114 seconds). We notice that the simulation time increases approximately linearly as the time duration of the source file increases. Thus, the simulation efficiency can become a significant factor when we try to quickly detect intrusions that involve large data files.

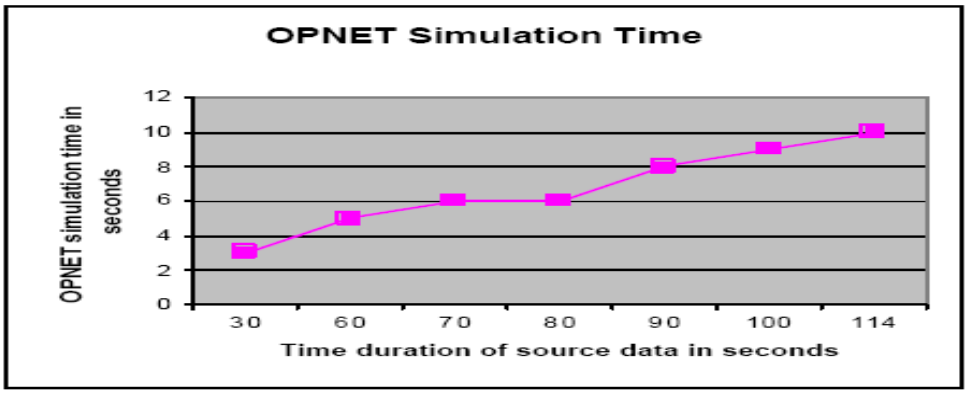

Fig. 12: OPNET simulation time of the ProcessTable attack

V. CONCLUSION

Information security is of great concern these days due to the activities of hackers and malicious users on the Internet. Securing information has become a critical issue and is of growing concern as computer systems worldwide become increasingly vulnerable to the rapid increase in the volume of information being transmitted across networks and over the Internet. In this paper, we proposed a technique that provides a robust intrusion detection system against attackers. We discussed the design and implementation of the new tool that was developed for this paper. We demonstrated that the tool provides a robust web-server security against IDS - based attacks. The tool contains crucial IDPS components and provides capabilities that other tools failed to provide. The features include IDPS management configuration interface, ability to block the attackers, and automatic email notification. The technique puts scalability and appropriate security in mind. The framework is made up of a Network Intrusion Detection System (NIDS) for detection of traffic to and from a given network or sub network, a Host based Intrusion Detection System (HIDS) and a line for possible Intrusion Prevention Systems (IPS). The technique is implemented by modeling network using OPNET, a network simulation software; since a real life implementation is very costly. Our result shows that the technique provides an intrusion detection system that can be used to monitor user's activities on the Internet and other networks with relatively minimal false alarms.

REFERENCES

[1] Damien, H. and Mathew, W. (2003). Security for Internet Banking: a Framework, *Logistics Information Management Vol.16, No. 1*, pp. 64-73.

[2] Joseph, S. and Rod, A. (2003). Intrusion Detection: Methods and Systems. Part II. *Information Management and Computer Security Vol. 11, No. 5*, pp. 222-229.

[3] Arafat, H. (2001). A New Model for Monitoring Intrusion Based on Petri Net, *Information Management and Computer Security, Vol. 9, No. 4*, pp. 175-182.

[4] Chan, P. and Wei, V. (2002). Preemptive Distributed Intrusion Detection Using Mobile Agents. Paper Presented at *IEEE International Workshop on Enabling Technologies*, June 2002, Carnegie Mellon University, Petersburg, P.A.

[5] Rod, H. Darren, M. and Hai, T. (1999). An Introduction to Automated Intrusion Detection Approaches. Information Management and Computer, 1999.

[6] Zhou, J., Carlson, A. and Bishop, M. (2005). Verify Results of Network Intrusion Alerts Using Lightweight Protocol Analysis, *Proc. of the 21st Annual Computer Security and Applications Conference*, 2005, (ACSAC 2005).

[7] Elhenawy, I., Raid, A. E.-D., Hassan, A. Awadallah, N. (2011). Visualization Techniques for Intrusion Detection – A Survey. *International Journal of Computer Science & Engineering Survey (IJCSES)*, Vol. 2, No. 3, August 2011, 107-119.

[8] Hwang, K. Cai, M. Che, Y. and Qin, M. (2007). Hybrid Intrusion Detection with Weighted Signature Generation Over Anomalous Internet Episodes, *IEEE Transactions on Dependable Computing, Vol. 4, No. 1*, pp. 41-55.

,[9] Ajith, A. Crina, G. and Yuehui, C. (2001). Cyber Security and the Evolution of Intrusion Detection S*ystems. Information Management and Computer Security 9(4)*, pp. 175-182.

[10] Ye, N. and Chen, Q. (2001). Profile-Based Information Fusion for Intrusion Detection. *Proc. of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy*, Wets point NY, June 5-6, pp. 227-230.

[11] Daniels, T. E. and Spafford, E. H. (1999). Identification of Host Audit Data to Detect Attacks on Low-Level IP Vulnerabilities. *Journal of Computer Security Vol. 7, No. 1*, 1999, pp. 3-35.

[12] Ebrahim, H. E. Badr, S. M.and Shaheen, M. A. (2012). Phases vs. Levels using decision trees for intrusion detection system, *International Journal of Computer Science and Information Security (IJCSIS),Vol. 10, No. 8*, pp. 1-7.

[13] Ning, P. Cui, Y. Reeves, D. S. and Xu , D. (2004). Technique and Tools for Analyzing Intrusion Alerts, ACM Journal, Vol. v, No. N, pp. 1-44.

[14] Faizal, M. A , Mohd, Z. M., Shahrin, S. Robiah, Y. Siti, R. S. and Nazrulazhar, B. (2009). Threshold Verification technique for Network Intrusion Detection System, *International Journal of Computer Science and Information Security, 2(1),* 2009, pp. 1-8.

[15] Wang, J. and Lee, T. (2002). Enhanced Intranet Management in DHCP-Enabled Environment, *Proc. of the 26th Annual International Computer Software and Applications Conference 2002, COMPSAC'02.*

[16] Sodiya, A. and Akinwale, A. (2004). A New Two - Tiered Strategy to Intrusion Detection, *Information Management and Computer Security, Vol. 12, No. 1*, 2004, 27-44.

[17] Steve, H. David, C. Yen, Y. and x David, Y. (2000). Awareness and Challenges of Internet Security, *Information Management and Computer Security, Vol. 8. No. 3*, pp. 131-143.