Article 3

Intrusion Detection System (IDS) : Investigating Snort Performance in Windows and Ubuntu due to Flooding Attack

Abidah Mat Taib, Nur Syahirah Shayuthi,
Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA Perlis Branch, Malaysia

Abstract
Intrusion detection is an important technology that can help in managing threats and vulnerabilities in this changing environment. Computer technology is more and more ubiquitous, the penetration of computer in society is a welcome step towards modernization but society needs to be better equipped with challenges associated with technology. Thus, with the help of intrusion detection system (IDS) that can be used to monitor network for any attack and intrusion, it can reduce the security issues and help people to curb with the advance threat. This project aims to provide insight to small organization, employee and student to have a secure environment in their personal computer. The objectives of this project is to set up an isolate local area network (LAN) to imitate a real network environment using Graphical Network Simulator-3 (GNS3) and to create the scenario for analyzing Snort IDS performance in Windows and Ubuntu due to flooding attack. Basically, this project uses a router in GNS3 that can act as a real router. The IDS was implemented on the PC1 while PC2 acts as an attacker that send a flooding attack to PC 1. The timer was set for 2 minutes and the performance was analyzed based on drop packet and throughput. The result shows that the performance of Snort is better in Ubuntu compared to Windows in term of its drop packet and throughput.

Keywords: *Intrusion Detection System, Snort, GNS3, performance analysis, flooding attack*

Introduction

Internet is becoming very important in people's lives since it can be a medium for them to communicate with others easily at a lower cost. Furthermore, people usually use internet as a medium to share files, get entertainment, search for information and do other activities that give benefits to them (Muniandy, 2010). However, not all things connected to the internet give advantages to the users. People who are connected to the internet have their own intentions and make choices whether to do good or bad things.

There are many types of computer security risk that can cause damage to personal computers such as internet and network attacks, unauthorized access and use, hardware theft, software theft, information theft and system failure (Ahmad, 2012). So, it is important for all users to give attention about computer security. There are various ways to secure and defend the system from unauthorized use for example encryption, use of a firewall, anti-virus software and intrusion detection (Debra & Shinder, 2006). Intrusion detection system (IDS) can provide security services using many conventions or patterns and it can provide robust, highly flexible, portable and fully controlled protection against an entire field of threats (Jeganathan & Prakasam, 2014). Snort is an open source IDS that can be freely installed in various OS.Kuldeep, Tyagi and Richa (2014) have implemented Snort IDS in cloud environment to deal with pretense attacks.

Problems happen when people are not concerned about the security issues while using the internet and they do not know how to secure their own personal computers ("Computer Threats | Monster.com," 2017). In common, users either use Windows, Ubuntu or other operating systems (OS). As reported by 3schools (2013), most users prefer to use Windows OS with usage percentage of 77.3%, Linux OS with 5.5% and other operating system with 6.4%. Thus, knowing how Snort on Windows and Snort on Ubuntu operate is important. Hence, this project aimed to investigate and give understanding about installing and applying IDS in a local network and observing how a host detects an intrusion. The observation was focused on the performance of a computer running IDS on Windows and Linux operating system towards denial of service (DoS) attack.

This paper aims to provide insights about the importance of detecting intrusion and how Snort can be used for that purpose. Thus, investigation of Snort implementation in Windows and Ubuntu was carried out where respective testing and experimentation had been set up on an isolated network using a Graphical Network Simulator-3 (GNS3) (GNS3 2016). The remainder of this paper is organized as follows. Next section will be presenting the overview of DoS and IDSs and where they are used. It follows with explaining the methodology and experiment setup, result and analysis, and finally the conclusion.

Overview of DoS Attack

Flooding attack is a common DoS attack that is intended to bring a system or service down by flooding the system with a large amount of data (Manna & Amphawan, 2012). Protecting a network against distributed DoS attack is quite a challenge. Nevertheless, a possible effort can be taken by applying appropriate IDS rules to guard against flooding attack. It is one of the mechanisms that is likely capable of defending DoS attack at the early stage. Some well known flooding attacks are related to Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) (Li, Li, & Zhao, 2009). The flood attack of TCP SYN or TCP ACK is a very common attack. TCP SYN flood is a type of DoS attack in which an attacker sends a progression of SYN requests to a target's system and tries to consume enough server resources to make the system unresponsive to genuine activity (Bogdanoski, Shuminoski, & Risteski, 2013).

The attack starts as an ordinary TCP connection in which the client and server exchange data in TCP packets. The attacker can craft a huge number of SYN packets with spoofed source IP addresses that represent TCP client keeps on sending SYN packets to the server, these SYN packets tell the server that a connection is requested. The server consequently reacts to each client with an ACK packet. The client is supposed to react with another ACK packet accepting the connection in order to set up the session. The server holds these sessions open, anticipating the last packet in the sequence. In this attack, no response ACK packets from the clients arrive at the server. This scenario makes the server fills up the accessible connection and denies any request of client access (Anand, 2012).

UDP utilizes a basic transmission model without implicit handshaking dialogues for providing dependability, requesting, or data integrity. Therefore, UDP gives an unreliable service and datagram may arrive out of order, show duplicate, or missing without notice. So, the user application (program) responsible of taking care UDP assumes that error checking is either necessary or performed, to avoid from the overhead of such handling at the network interface level.

Furthermore, in UDP flood attack, it is similar to Internet Control Message Protocol (ICMP) flood attack, it sends a substantial number of UDP messages to the target in a short time, so that the target will be busy responding and transmitting the normal data packets (Kumar & Rai, 2012).

Overview of IDS

IDS is the process of monitoring the events occurring in a computer or networked system and analyzing events for signs of possible incidents which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices (Kumar et al., 2013). IDS can be in the form of a software application that operates based on misused (signature-based) detection or anomaly detection (Buchanan, 2011). An IDS can be placed within the network to monitor network traffic, such as looking for known attacks or virus signature or it can be placed on hosts to detect an actual host intrusion. IDSs that monitor data packets on the network and try to determine an intrusion based on network traffic is known as network IDSs (NIDS). NIDS can run on a host or on network-based. Snort (Orebaugh, Biles & Babbin, 2005) is an open source NIDS that is freely available. Honeypots (Buchanan, 2011) also is an example of IDS that is used to attract an intruder and detect the intrusion at the early stage.

This project focuses on Snort since it is widely used and can detect both signature- and anomaly-based detection. Snort runs as a background process and reads-in a set of rules and monitors the network traffic to produce event data and a log (Buchanan, 2011). In order to learn the application of host-based IDS, an experiment test bed was setup to investigate Snort performance in detecting DoS attack in Windows and Ubuntu. The respective experiment was conducted in an isolated network that was built using GNS3. The testing involved is described in the methodology.

Methodology

The first step was getting the appropriate hardware and software ready for the setting up of the experiment testbed. Then, installation of the required software on the respective hardware was done. Upon installation finished, an isolated network was created using a GNS3. GNS3 allows us to design complex network topologies. It can run simulations or configure devices ranging from simple workstation to powerful Cisco routers. It is based on Dynamips, Pemu/Qemu and Dynagen(GNS3, 2016). To make sure the network is correctly configured and ready for testing, network connectivity test was carried out. Subsequently, four scenarios of experiment were conducted that aimed to investigate performance of Snort running on Windows and Ubuntu in detecting SYN flood and UDP flood attacks. Time was set for 2 minutes for all experiment scenarios. Each experiment was repeated three times.

Experiment setup

This testbed setup involves two PCs. First PC was operated by Windows OS and installed with a GNS3 version 1.5.3. This PC also was installed with Virtual Box that also runs another guest OS, Ubuntu 12.04 that acted as an attacker. Another PC (a laptop) was installed with dual stacked OS: Windows 7.0 and Ubuntu 12.04. This PC acted as a victim. Figure 1 shows the experiment testbed. An isolate network was setup by configuring two routers that are connected to PC1 and PC2 respectively. They represent two different networks that are connected in the cloud. Snort 2.9.9.0 was installed as an IDS on the victim machine. TCP and UDP flooding attacks were launched using an open source attacking tool, Hping3 (Sectools.org). A packet analysis software, Wireshark

(Sanders, 2007), was also set to run on both attacker and victim to monitor the behavior of the network.

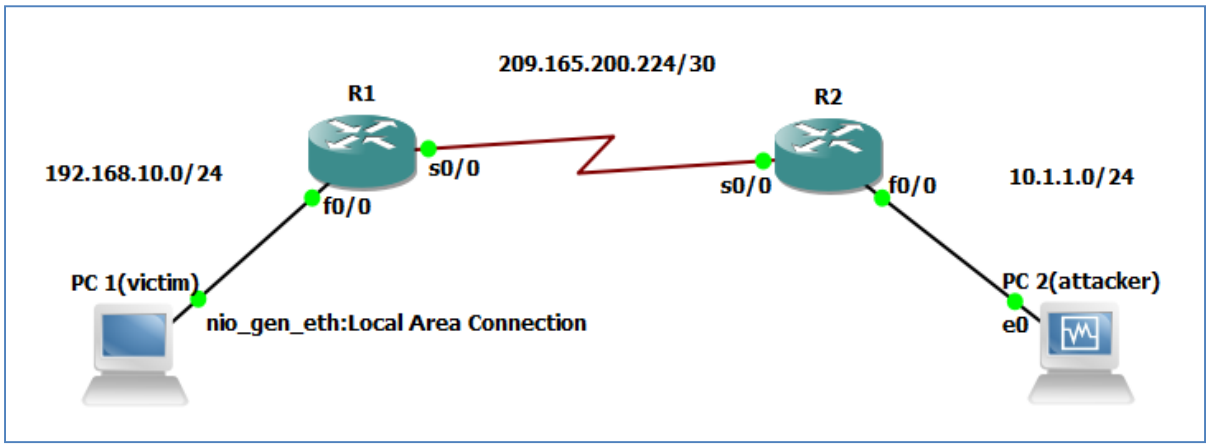

Figure 1: Experiment Test bed

Studying the performance involved creating a valid and reliable record of performance by means of systematic observations that can be analyzed with a view to facilitating change (Alhomoud, Munir, Disso, Awan, & Al-Dhelaan, 2011). The performance metric used in this project are throughput and packet input /output total that consist of the results for drop packet.

Throughput is a measure of how many number of units of data that a system can handle in a given measure of time (Wuu, Hung, & Chen, 2007). It is connected comprehensively to the system that running from different parts of PC and network system to associations. Related measures of system profitability incorporate the speed with which some particular workload can be finished, and reaction time, the measure of time between a solitary intuitive client demand and receipt of the reaction. The basic output in IDS include the timing statistics. In timing statistic, it includes the total second and packets as well as packet processing rates. This output is used to measure the throughput (Liao, Richard Lin, Lin, & Tung, 2012).

Packet input/ output totals in IDS can be measured after setting up the packet size. Then, the performance of IDS can be measured after the dropped packet segment shows its percentage (Rani & Singh, 2012).

A work by Alhomoud, Munir, Disso, Awan and Al-Dhelaan (2011) has tested and analyzed the performance of Snort and Suricata. They implemented both IDS on three different platform which are ESXi virtual server, Linux 2.6 and FreeBSD. They created three different scenarios with a different packet size and speed to analyze the performance of IDS. The result stated is based on dropped packet in each platform. In contrast, this work concerns only on Snort performance in Windows and Ubuntu platform while detecting SYN flood and UDP flood attacks.

Results and Analysis

Wireshark was activated to capture activities in Windows and Ubuntu before and after the attack was launched. Thus, it can demonstrate whether the attack was successful or not. In the experiment, attacker launched flooding attack to the victim for 2 minutes. Before the attack, no activities were spotted in the Wireshark display. During the occurrence of attack, Wireshark screen shows

flooding of packets from many sources targeting the victim. These are presented in Figure 2 and Figure 3.

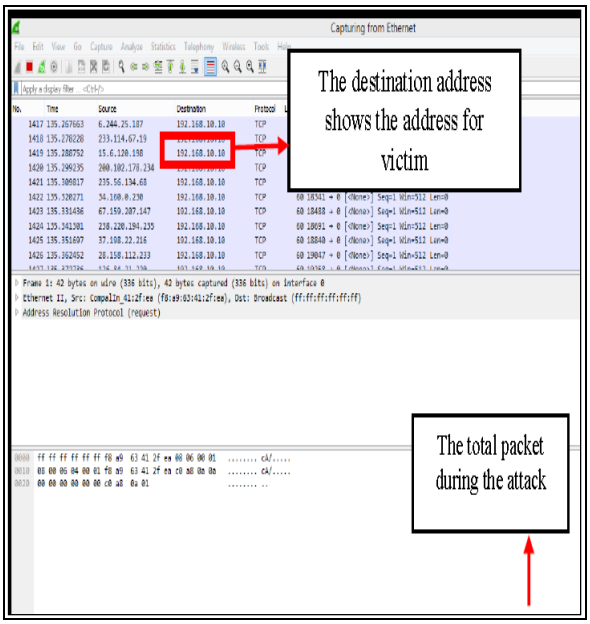

Figure 2: Wireshark during the attack in Windows

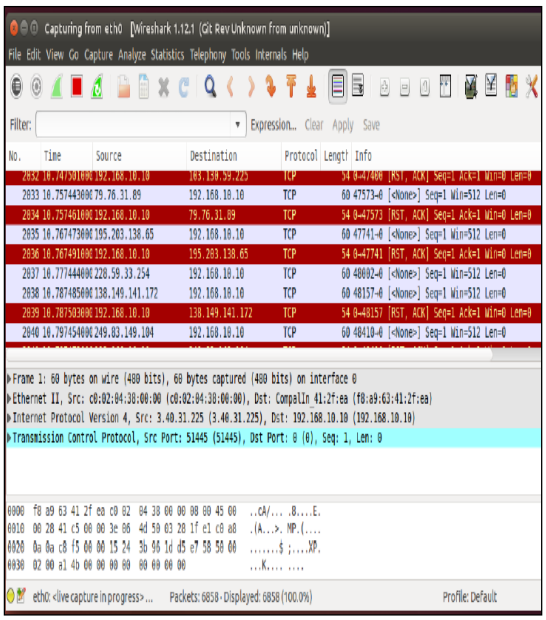

Figure 3:Wireshark during the attack in Ubuntu

i.Performance Analysis

Performance analysis of Snort IDS in Windows and Ubuntu are based on its dropped packet and throughput. During the attack, Snort was running until the set time is finished. Table 1 shows the overall results obtained from the experiments.

Table 1: Overall results

	Flooding attack	Windows			Ubuntu		
		Exp 1	Exp 2	Exp 3	Exp 1	Exp 2	Exp 3
Drop packet (%)	TCP	32.226	32.544	36.379	32.087	29.992	34.756
	UDP	35.06	34.934	34.524	27.075	27.332	29.534
Throughput (packet/min)	TCP	58146	59455	57968	106326	106445	106634
	UDP	65105	64821	67442	102432	102876	105066

For drop packet, either Snort on Windows or Snort on Ubuntu that has lower drop packet will be considered as better performance. While for throughput, Snort in which operating system that captured higher packet during the attack is considered as better performance. As can be noticed in Table 1, the value of drop packet in Ubuntu is smaller than in Windows. While for the observation on throughput, it can be seen clearly that the value of packet captured per minute for throughput in Ubuntu is higher than Windows. Figure 4and Figure 5 present the performance of Snort in term of its throughput on Windows and Ubuntu for TCP SYN flood and UDP flood respectively. Generally, if the packet captured per minute is higher, it means that the performance of snort is better in that platform. In order to confirm, further testing was implemented by conducting T test on both results.

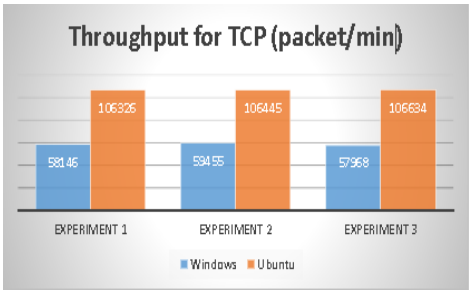

Figure 4:Throughput for TCP Flood

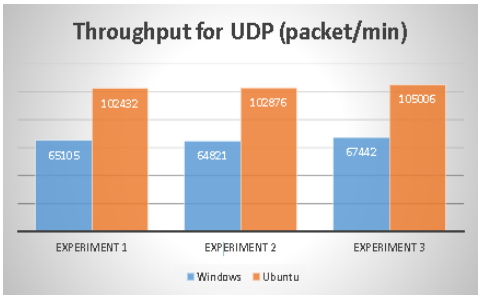

Figure 5:Throughput for UDP Flood

ii. T-Test Result

In order to determine whether the result is significant or not to the difference between the performance of Windows and Ubuntu in terms of dropped packet and throughput, independent T-test had been carried out.T-test is one of the analytical statistic to test the difference between the sample with variances of two unknown normal distributions. Basically, in this experiment the two sample that is used are Windows and Ubuntu whereas the two unknown normal distribution are dropped packet and throughput. Figure 6 shows the T-test results of drop packet for TCP SYN flooding attack.

T-Test

[DataSet1]

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	windows	33.71633	3	2.311412	1.334494
	ubuntu	32.27833	3	2.387756	1.378572

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	windows & ubuntu	3	.866	.333

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	windows - ubuntu	1.438000	1.217091	.702688	-1.585422	4.461422	2.046	2	.177

Figure 6: Drop packet for TCP SYN flooding attack

The result shows that there is no significant different between the performance of Snort IDS on Ubuntu and Windows in term of drop packet due to SYN flooding attack, as the ($p > 0.05$). That means, the performance of Snort IDS in both Windows and Ubuntu is similar. On the other hand, the T-test result for drop packet in Windows and Ubuntu during UDP flooding attack shows that the ($p < 0.05$). This means, the different of performance of Snort IDS in Windows and Ubuntu in term of drop packet is significance. Thus, it can be concluded that Snort IDS on Ubuntu has better performance in terms of its drop packet when dealing with UDP flooding attack.

As for throughput in Windows and Ubuntu during the TCP flooding attack, the result of T-test indicates that the performance of Windows and Ubuntu in term of throughput is significance as the ($p < 0.05$). Therefore, it can be concluded that Snort IDS on Ubuntu has better performance in term of throughput while dealing with TCP flooding attack. The similar T-test result ($p < 0.05$) also was showed for throughput in Windows and Ubuntu during the UDP flooding. It indicates that there is a significance different. Thus, it can be concluded that Snort IDS on Ubuntu has better performance in terms of throughput for UDP flooding attack.

Conclusion

This paper has described IDS and how it can be applied for attack detection. Although the Snort IDS only was installed in two OS: Ubuntu and Windows and tested against two types of flooding attack, readers can acquire some fruitful information related to installation and experiment setup. Users can gain some knowledge from description of the experiment in which the paper has demonstrated the tests and discussed the results accordingly. As discussed in the result, Snort on Ubuntu is most likely has better performance in term of drop packet and throughput compared to Windows. In future, more testing of attacks can be conducted to observe the performance of IDS.

References

3schools. (2013). OS Statistics. Retrieved June 10, 2017, from https://www.w3schools.com/browsers/browsers_os.asp

Ahmad, A. (2012). Type of Security Threats and It 's Prevention, *3*(2), 750–752.

Alhomoud, A., Munir, R., Disso, J. P., Awan, I., & Al-Dhelaan, A. (2011).Performance evaluation study of Intrusion Detection Systems.*Procedia Computer Science*, *5*, 173–180. https://doi.org/10.1016/j.procs.2011.07.024

Anand, A. (2012). An Overview on Intrusion Detection System and Types of Attacks It Can Detect Considering Different Protocols. *International Journal of Advanced Research in Computer Science and Software Engineering*, *2*(8), 94–98.

Bogdanoski, M., Shuminoski, T., & Risteski, A. (2013). Analysis of the SYN Flood DoS Attack. *I.J. Computer Network and Information Security*, *8*(8), 1–11. https://doi.org/10.5815/ijcnis.2013.08.01

Buchanan, W. J. (2011). *Introduction to Security and Network Forensics*, USA :CRC Press.

Computer Threats | Monster.com. (2017). Retrieved June 16, 2017, from https://hiring.monster.com/hr/hr-best-practices/monster-training/security-center/avoid-computer-threats.aspx

Debra, B., &Shinder, L. (2006). 10 things you can do to protect your data. Retrieved June 16, 2017, from http://www.techrepublic.com/article/10-things-you-can-do-to-protect-your-data/

GNS3 | The Network Journal. (2016). Retrieved March 15, 2017, from https://cyruslab.net/tag/gns3/

Jeganathan, I. T. V. S., &Prakasam, A. (2014). Secure the Cloud Computing Environment from Attackers using Intrusion Detection System. *International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014), 2(2), 181–186.*

Muniandy, B. (2010). Academic Use of Internet among Undergraduate Students : A Preliminary Case Study in a Malaysian University. *International Journal of Cyber Society Education*, *3*(2), 171–178.

Kuldeep, T., Tyagi, S. S., &Richa, A. (2014). Overview - Snort Intrusion Detection System in Cloud Environment, *4*(3), 329–334.

Kumar, B. S., Sekhara, T. C., Raju, P., Ratnakar, M., Baba, S. D., &Sudhakar, N. (2013). Intrusion Detection System- Types and Prevention, *4*(1), 77–82.

Kumar, S., & Rai, S. (2012). Survey on Transport Layer Protocols : TCP

Li, M., Li, J., & Zhao, W. (2009). Experimental study of DDOS attacking of flood type based on NS2. *International Journal of Electronics and Computers*, *1*(500), 143–152. Retrieved from http://www.umac.mo/rectors_office/docs/weizhao_cv/pub_refereed_journals/2009_ref_journals/IJEC_Dec 2009.pdf

Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., & Tung, K.-Y. (2012). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

Manna, M. E., & Amphawan, A. (2012). Review of Syn-Flooding Attack Detection Mechanism. *International Journal of Distributed and Parallel Systems (IJDPS)*, *3*(1), 1–19. https://doi.org/10.5121/ijdps.2012.3108

Orebaugh, A., Biles, S. & Babbin, J. (2005). *Snort Cookbook*, USA: O'Really.

Rani, S., & Singh, V. (2012). SNORT: An Open Source Network Security Tool for Intrusion Detection in Campus Network Environment. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(1), 137–142. Retrieved from http://www.ijctee.org/files/VOLUME2ISSUE1/IJCTEE_0212_24.pdf

Reddy, G. N., & Reddy, G. J. U. (2014). A Study of Cyber Security Challenges and Its Emerging Trends on Latest Technologies. *International Journal of Engineering and Technology*, *4*(1), 48–51.

Sanders, C. *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems.* (2007). No Starch Press, USA: San Francisco.

SecTools.Org. (2017), *Hping3*.Retrieved March 10, 2017 from http://sectools.org/tool/hping/

Varga, A., & Hornig, R. (2008). An Overview of the OMNeT++ Simulation Environment. *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 60:1--60:10. https://doi.org/10.4108/ICST.SIMUTOOLS2008.3027

Wuu, L. C., Hung, C. H., & Chen, S. F. (2007).Building intrusion pattern miner for Snort network intrusion detection system.*Journal of Systems and Software*, *80*(10), 1699–1715. https://doi.org/10.1016/j.jss.2006.12.546