Threshold verification using Statistical Approach for Fast Attack Detection

Mohd Faizal Abdollah[1], Mohd Zaki Mas'ud[2], Shahrin Sahib@Sahibuddin[3], Robiah Yusof[4], Siti Rahayu Selamat[5]

Faculty of Information and Communication Technology
Univeristi Teknikal Malaysia, Karung Berkunci 1200, 75450 Ayer Keroh, Melaka.
Tel : 06-2332510, Fax : 06-2332508
{[1]faizalabdollah,[2] zaki.masud, [3]shahrinsahib, [4]robiah, [5]sitirahayu}@utem.edu.my

ABSTRACT

Network has grows to a mammoth size and becoming more complex, thus exposing the services it offers towards multiple types of intrusion vulnerabilities. One method to overcome intrusion is by introducing Intrusion Detection System (IDS) for detecting the threat before it can damage the network resources. IDS have the ability to analyze network traffic and recognize incoming and on-going network attack. In detecting intrusion attack, Information gathering on such activity can be classified into fast attack and slow attack. Yet, majority of the current intrusion detection systems do not have the ability to differentiate between these two types of attacks. Early detection of fast attack is very useful in a real time environment; in which it can help the targeted network from further intrusion that could let the intruder to gain access to the vulnerable machine. To address this challenge, this paper introduces a fast attack detection framework that set a threshold value to differentiate between the normal network traffic and abnormal network traffic on the victim perspective. The threshold value is abstract with the help of suitable set of feature used to detect the anomaly in the network. By introducing the threshold value, anomaly based detection can build a complete profile to detect any intrusion threat as well as at the same time reducing it false alarm alert.

Keywords: *Intrusion detection system, fast attack, Statistical Process control.*

1.0 INTRODUCTION

Since the development of internet in 1969 networks have grown in both size and importance. The Internet which is also known as the TCP/IP networks has become the mean to share and distribute information. It also can be exploited to attack a host. Nowadays, attacks are becoming easier to launch as the tool is easily obtained and the attack itself has becoming more sophisticated. In order to overcome the attack, one needs to understand the phases an attacker take to launched it activity. Intrusion attack involved five phases which are reconnaissance, scanning, gaining access, maintaining access and covering tracks (CEH, 2005).

The first two phases are considered as the initial stage of an attack, whereby both of the phases is used to collect information on the vulnerability of the targeted machine. The information gain can be either be the services offer on the network, the port open and type of operating system used. The result of the finding will help the attacker to decide what tool to use in the next phase of the attack. These phases can be classified into two categories which are fast attack and slow attack. Fast attack is defined as an attack that uses a large amount of packet or connection within a few seconds (Lazarevic et al, 2003) and the slow attack is considered as an attack that takes a few minutes or a few hours to complete. Existing detection system such as Snort (Softfire inc., 2007) and Bro (V.Paxson, 1999) combine fast attack and slow attack detection into one module, this may cause late detection especially for the fast attack. Early detection is needed to prevent a more serious damage to a vulnerable network, by reducing the further step in attacks; the losses due to the security breach can be minimized.

This paper presents a novel framework on detecting fast attack that focusing on the number of connection made by an attacker towards a single victim at the initial stage of the attack. By focusing on this information, administrators will have valuable information regarding the level of security of the compromised machine. Therefore investigating and make necessary action is a must to secure the machine from future attack.

The rest of the paper is structured as follows. Section 2 discuses the background of fast attack framework, Section 3 presents the methodologies and the technique use in creating the fast attack module. Section 4 elaborates on the result validation. Finally, section 5 conclude and discuss the future directions of this work.

2.0 BACKGROUND

Similar worked on fast attack has been done but most are concentrating on detecting fast attack based on specific attack such as worm, scanning and Denial of service (Dos) or Distributed Dos (DDos). Worked done by Xio et al. (2005) only consider TCP-SYN flag in their investigation

to detect DDos attack. Whereas Zou et al (2006) used Kalman Filter estimation algorithm in detecting code red and blaster worm using simulation approach. Robertson et al. (2003) and Bro (V.Paxson, 1999) both are focusing on the failed connection attempts. Whereas snort uses portscan2 scan-detection preprocessor, which take into account distinct connection rather than TCP-SYN packet (Jung et. al. 2004). Hence, it may generate a lot of misclassification in a case of a single host sends multiple SYNs in the same failed connection attempts.

2.1 Fast Attack Framework

By analyzing the nature of an attack, criteria such as number of new connection and derived features can be used to detect the fast attack. Shahrin et. al.(2007a) have proposed the fast attack detection novel framework as depicted in figure 1.

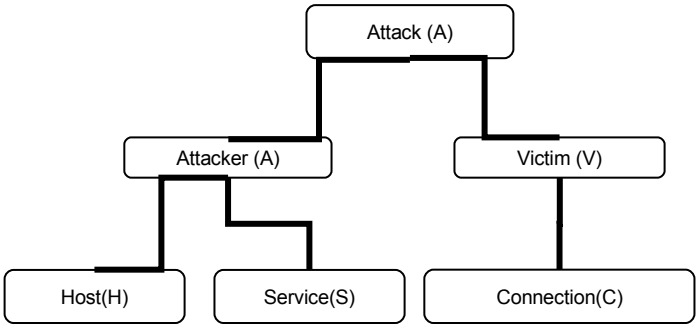


Figure 1: Framework of Fast attack detection

From the framework the attack is divided into 2 categories, which are Attacker and Victim. In the attacker perspective, the attack is caused by a single host and to make it more reliable it is divided into 2 subcategories to make it more reliable. The host subcategories or AAH is to detect any attack originated from single host that targeted multiple hosts such as scanning attack. Meanwhile the subcategories Service or AAS is referring to detection of an attack towards the services offered by victim. This paper will concentrate on the Victim perspective, in which it focuses on number of host connected to the single destination host. As a result, any compromise from any host to the victim (destination) may be detected.

2.2 Features selection

The success of an Intrusion Detection System depends on the decision upon a set of features that the system is going to use for detecting the attacker especially the fast attacks. Different researchers use different names for the same subset of feature while others use the same name but different types (Onut and Ghorbani, 2006). Understanding the relationship as well as the influence of the features in detecting the fast attack is necessary to avoid any redundant features selected for the intrusion detection system. Shahrin et al. (2007a) constructed a minimum set of features by integrating the new features and features from the KDDCUP99 (1999), in order to detect the fast attack from the perspective of the victim. Table 1 describe the features involve in this research.

Table 1: *Feature selection*

Feature	Description	Category
Timestamp	Time the packet was send	Basic Features
Duration	Duration of connection.	
IP	Addresses of host	
Protocol type	Connection protocol (e.g tcp)	
Flag	Status flag of the connection	
Service	Source and Destination services.	
Dest_count	Number of connection having the same destination host (AVC).	Derived Features

Dst_count is generated using the frequency episode technique as this technique can discover what time-based sequence of audit events frequently occur together (Lee and Stolfo, 2004). Analyzing the relationship between features is also important to reduce the selection of the features because some features may cause or contain negative correlation (Chebrolu, Abraham and Thomas, 2004). This research found that by using these features, the system is still capable of detecting fast attack with a minimum false alarm. Furthermore, it also can speed up the detection time especially in detecting the fast attack.

2.3 Threshold analysis

There are two techniques that can be used in selecting the appropriate threshold to distinguish between the normal network traffic and abnormal network traffic. The techniques are static threshold value and dynamic threshold value. Dynamic threshold value requires training or priori knowledge of the network activity before the threshold is selected (Idika and Mathur, 2007). Generating priori knowledge require human expert and is time consuming since network traffic contains huge amount of information and the complexity involves protocol which requires an in depth knowledge of human expert to analyze it (Mellia et al, 2006). Therefore, this research will focus on static threshold value because selecting static threshold is very useful to prevent the intrusion activity before the attacker begins to launch the attack (Idika and Mathur, 2007).

Moreover, static threshold has been widely used by commercial products in detecting the attacker in the network. Netscreen IDS developed by Juniper (Juniper, 2007) also adopted the static threshold inside the configuration to tackle the Denial of Service activity.

Commercial IDS, the most popular intrusion detection software such as Bro (Bro, 2007) and Snort (Softfire Inc., 2007) still uses the static threshold mechanism to identify the attacker. As stated earlier in this paper, the drawback of these products is the implementation of both the fast attack and slow attack in one module which can slow the detection process.
Other research such as Gates and Damon (2005), Ye and Chen (2001) and Leckie et al, (2002) also used static threshold mechanism in identifying the attacker. However most of them are focuses more on the individual attack rather than the behavior attacks and the analysis only concentrates on the detection technique rather than revealing the feature influence. Furthermore, none of the previous researcher proposed a proper technique to identify the static threshold. Most of them used observation technique without asserting a clear explanation on the selection of the static threshold using the observation technique. To overcome all these weaknesses, this research determined the static threshold using the observation technique combined with experiment to identify the appropriate threshold for the victim category. The result of the observation technique is then validated using Statistical Process Control (SPC).

3.0 IMPLEMENTATION

This section will describe the methodology, techniques and the statistical approach used. Python Software Foundation (2007) and shell scripting is used to develop the platform and the underlying operation system is Linux. The system was designed in such a way so that it can work and extract feature under real time traffic environment.

3.1 Methodology

The main module for the fast attack detection is depicted Figure 2. The traffic is captured by TCPdump (2007) and the information from the tcpdump is processed through the Feature Extraction, Time Based, and Threshold Detection module.

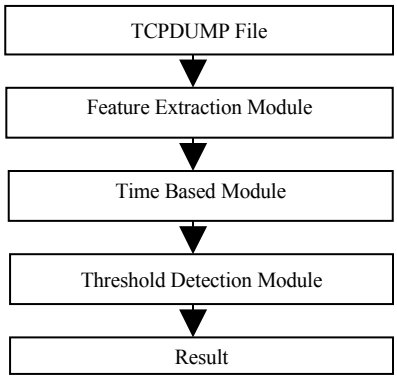


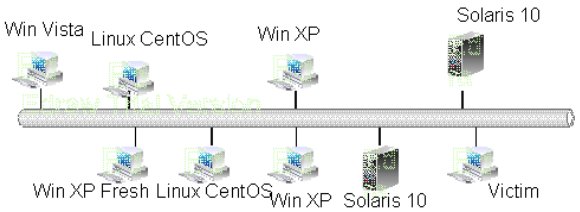
Figure 2: Fast attack main module.

The Feature extraction module separated the file into their respective protocol such as TCP, UDP and ICMP. For TCP protocol, this project excluded services from port 80, 443, 135 until 139. Next, the module sends the data as a continuous stream to Time Based Module. Time Based Module calculate the statistic of the frequent event occur between 1 second time interval for each number of host connected to a single destination host.

The next module is the threshold module, which is the backbone of the fast attack detection from the victim perspective. Choosing the right threshold is very important to this research, selecting too low, may generate excessive false positive while too high, will cause the system to miss less aggressive scanner. The next section will discuss on the techniques used by this research in determining the threshold value. The output of this module determined the real traffic and the anomalous traffic.

3.2 Observation and Experiment Techniques

The observation is done on real network traffic from one of the government agency site. The purpose of the observation is to identify the average connection made by host or hosts to single victim within one second time interval. By doing this, the connection made to single victim can be identified. The result from the observation technique will be compared to select the appropriate static threshold for the fast attack detection. Meanwhile, for the experiment technique, a small local area network (LAN) has been setup, refer figure 3. The experiment consists of multiple operating systems such as Windows XP Professional Service Pack 2, Windows Vista and Linux



CentOS Version 4.4.

Figure 3: Experimental setup

The purpose of the experiment is to identify the normal connection made by each of operating system. By doing this, the normal behavior of host in transmitting network packet to the destination host within one second time interval is identified. The result from each of the operating system will be captured and compared with each other. Another comparison process will be done between the result from the observation and result from the experiment. The comparison result will be used to select an appropriate threshold for the fast attack detection. After selecting the threshold, the verification process using the SPC model

will be done to validate the threshold selection from the observation and experiment before using it inside the real time environment.

3.3 Statistical Process Control

SPC is a powerful tool for achieving process stability and improving capability through the reduction of variability. The major objective of a statistical control chart is quick to detect the occurrence of assignable causes or process shifts so that investigation of the process and corrective action may be undertaken before many nonconforming units are manufactured (Montgomery and George, 2007). Due to the fast attack which needs to be detected as quickly as possible, therefore a statistical control chart is deemed suitable used in this research. Furthermore, a statistical control chart can be used to verify the threshold selection before using the threshold in real time environment for detecting the fast attack.

The general model for the control chart is introduced by Dr. Walter A. Shewart (Montgomery and George, 2007). Let P be a sample statistic that measures the quality characteristic of interest, and the mean of P is μ and the standard deviation of P is σ .The center line, the upper control limit and lower control limit equation is stated in (1).

$$\text{UCL} = \mu + k\sigma$$
$$\text{CL} = \mu$$
$$\text{LCL} = \mu - k\sigma \qquad (1)$$

Where *k* is the distance of the control limit from the center line. The k value is also used by Ye et al, (2001) in their research in detecting the intrusion activity based on audit trail. In this research the value of k equal to 3 and is used for developing the Shewhart control chart.

The chart contain a center line (CL) that represent the average value of the quality characteristic corresponding to the in-control stated. The two others horizontal lines indicate the upper control limit (UCL) and lower control limit (LCL). If there are points which exceed the control limit, these indicate that the point is out of control. During the research the output from the time based module will be used to construct the SPC chart. The threshold value selected from the experiment and observation will be used as a mean for constructing the SPC chart. The mean value indicate the normal behavior of network traffic inside the network. If the number of connection to the single destination host exceeds the value of normal threshold, then it will recognize that there is an abnormal behavior in the network.

4.0 RESULT VALIDATION.

In this research, the observation and the experimental techniques are concentrated on port 21, 25, 53, 110, 135, 139, and 445. Port 25 and 110 has been selected because these ports are important for sending and receiving emails. Furthermore, both this ports are easily compromised by spamming activity and the incident cases from the spam activity are increases every year (MyCert, 2008). Port 53 called as Domain Name Service (DNS) is selected because this port is important to resolution process especially in determining the IP address to a name of a host (Nemeth et al, 2002). Protecting DNS is also important because any single failure to DNS root server or any DNS queries may result in catastrophic consequences (Wang et al, 2003). Port 21, 135, 139 and 445 is the most popular port that is used by worm activity to compromise the host (Cybersecurity, 2008). Therefore, it has been selected as one of the port for observation.

4.1 Observation and Experimental result.

The observation technique will identify the average number of connection made to single victim within time interval of one second. The observation of network traffic is based on the well known port which has been used inside this research. The reason behind the selection of the well known port has been discussed earlier. By identifying the average number of connection made to a single victim, the pattern of traffic made by host to private port on a single victim was discovered. This result may help to distinguish between the normal and abnormal behavior of the network traffic. Besides the observation technique, simple local area network was also setup to discover the pattern of network traffic for a host from different operating system. Understanding the pattern of network traffic from different operating system may help to distinguish the normal and abnormal connection made by single host to single victim within one second time interval.

The observation techniques encompass the mean, minimum connection per second and maximum number of connection made by hosts or host to a single victim. Table 2 shows the summary of the result taken from the real traffic Site, from the table, the average number of connection per second for each port falls within one connection per second. Only two ports exceed three connections per second which are port 135 and port 139. By referring the technical report, this site was flooded with worm generated from port 135 and 139 (Shahrin et al, 2007b). Therefore the number of connection per second for port 135 and port 139 to single victim can be reached to 127 connections per second and 155 connections per second for each port respectively. As a conclusion, one connection per second can be chosen as a temporary threshold for the observation techniques.

Table 2: *Average Connection for Site B*

Port Number	Mean of	Minimum	Maximum

	connection per second	connection per second	connection per second
21	1.33	1	2
25	1.02	1	2
53	1.05	1	3
110	1	1	1
135	30.52	1	127
139	28.71	1	155
445	1.14	1	3

The experiment was done by capturing the network traffic generated by each of the operating system as depicted in figure 3. The victim host installed with CentOS 4.4 and used to capture all the network traffic to the victim host. The network traffic for each of the operating system will be read using TCPdump application and the traffic flow will be identified. The traffic flow will be analyzed to identify the number of connection made by each operating system to a single destination host. By doing this, the normal behavior of host in transmitting network packet to the destination host within one second time interval is identified and the average value obtain from the experiment is 3. The experiment result is shown in table 3:

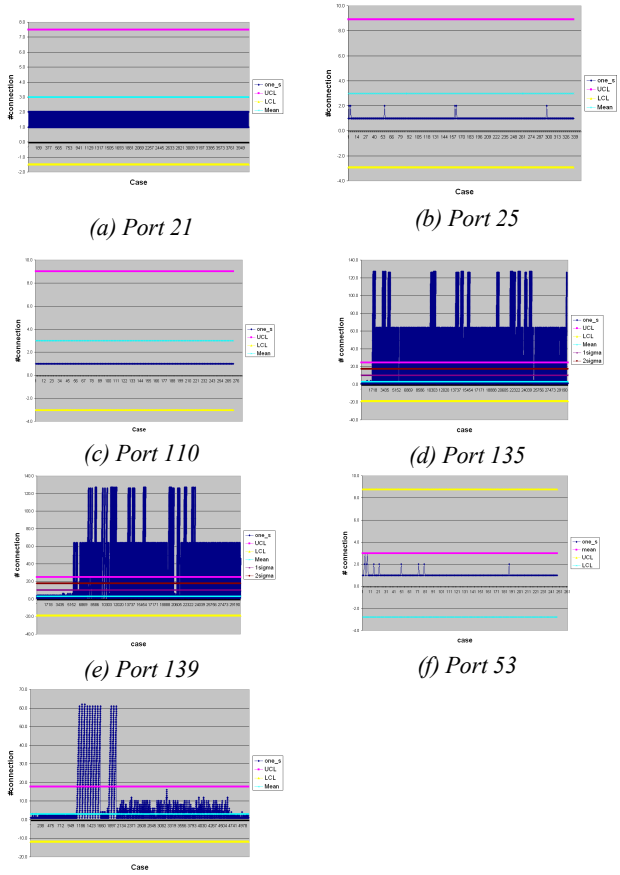
Table 3: *Experimental result*

Operating System	Number of Connection Per second
Windows XP Professionals Service Pack 2 (Fresh Install)	3
Windows Vista	3
Windows XP Professionals Service Pack 2	3
Linux CentOS 4.4	1
Solaris 10	1

From the observation technique result and the experimental result the average connection to a single host is in the range of 1 until 3 connections Therefore 3 connections per second can be considered as a normal connection and if the connection is more than 3 we can considered it as an abnormal behavior. Now the threshold selected can be verified using the Statistical Process Control.

4.2 Threshold Verification

The threshold verification is based on the data from the observation technique. The Shewhart Control Chart is contructed based on the well known port inside the observation site. The value for the Upper Control Limit (UCL) and Lower Control Limit will be computed based on the equation (8). The mean value used is referred to the normal threshold connection per second which is 3. It means that any connection that exceeds the normal threshold value is suspected as attack traffic. After the Shewhart Control Chart has been generated, any connection that exceeds the normal threshold will be examined manually to identify the accuracy of the selected threshold. If there is no connection per second which exceeds the normal threshold value, then it can be concluded that all the connections are following a normal behavior of the network traffic. From figure 4, port 21, 25, 53 and 110 shows a normal traffic as all the connections to this port is under the control limit and mean value. Whereas port 135, 139 and 445 have a connection over the control limit, based on the observation from the actual network traffic, the point is come from one of the host which generates excessive network traffic to multiple victim host on port 135, 139 and 445 and referring to the technical report done by (Shahrin et al, 2007b) on the observation site, shows that the hosts have been compromised by worm activity. There are other hosts which have been infected by worm and generated excessive network traffic to port 445. The total number of victims has been infected is 6 and the threshold managed to predict 5 out of 6. The false alarm generated at the victim is not a false alarm. It is because from the attacker perspective, the host who initiate the connection is considered as attacker because this host generated excessive network traffic to other victim. Based on the result, it shows that most of the host infected by worm will generate more than 3 packets per second. Therefore, the threshold selected may be useful to identify the worm activity as fast as possible since the objective of the research is to identify the fast attack.



(a) Port 21 *(b) Port 25*

(c) Port 110 *(d) Port 135*

(e) Port 139 *(f) Port 53*

(g) Port 445

Figure 4: SPC chart for port (a) 21, (b) 25, (c) 110, (d) 135, (e)

139, (f) 53, (g) 445.

5. 0 CONCLUSION AND FUTURE WORK.

The current work presents a framework in detecting fast attack from the victim perspective and introduced a minimum features that can be used to detect fast attack. In order to determine a threshold value that can be used to differentiate normal and abnormal network traffic, this new technique has incorporated observation technique combined with experiment and the finding is then verified using Statistical Process Control approach. Using a real network traffic data and data from the experiment setup, the research has introduced this new technique to be used to determine a threshold value to detect fast attack from the victim perspective.

For the future work, we would like to obtain normal network traffic behavior data based on network traffic simulation using the data from DARPA99. The result is then going to be used to validate the result obtains from the observation technique and experiment of this research. The next phase we will run the system on real-time network traffic to evaluate the effectiveness of the system in detecting fast attack in real time.

REFERENCES

B. Xiao, W. Chen, Y. He, E. H-M. Sha (2005), " *An Active Detecting Method Against SYN Flooding Attack*", In Proceeding 11th International Conference on Vol. 1, IEEE, 20-22 July 2005

Bro (2007), "*Bro Intrusion detection system*", Retrieved December 2, 2007, from http://www.bro-ids.org.

C. C. Zou, W. Gong, D. Towsley and L. Gao, (2006) *" The Monitoring and Early Detection of Internet Worms"* , In Proceeding IEEE Transactions on Networking, Vol. 13, 2006, 961-974.

Certified Ethical Hacker (CEH) Module 2005.

CyberSecurity Malaysia. (2008*). "E-Security Volume 14-(Q1/2008)". Technical Report for e-Security, CyberSecurity Malaysia"*, MOSTI, 2008.

Gates, C & Damon, B, Cpt. (2005). "*Host Anomalies from Network Data*". In Proceeding from the Sixth Annual IEEE SMC, June 15-17, 2005, USA.

Idika, N. & Mathur P. A. (2007). *"A Survey of Malware Detection Technique*". In Proceeding of Software Engineering Research Center Conference, SERC-TR286, February 25, 2007, USA.

J. Jung, V. Paxson, A. W. Berger and H. Balakrishnan, *" Fast Portscan Detection Using Sequential Hypothesis Testing",* In Proceeding of 2004 IEEE Symposium on security and Privacy, Oakland, CA, USA, May 2004.

Juniper Networks Inc. (2007), "*Junipe*r", Retrieve on 2 February 2008, from http://www.juniper.net

KDDCUP99 (1999), "*KDD CUP 99 dataset*", retrieve December 5, 2007, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Lawrence Berkeley National Laboratory (2008),*" Bro*", Retrieve on 2 January 2008, from http://www.bro-ids.org

Lazarevic A., Ertoz L., Kumar V., Ozgur A. & Srivastava J. (2003). *"A comparative Study of Anomaly Detection Schemes in Network Intrusion detection*". Proceeding of SIAM International Conference on Data Mining, May 1-3, 2003, USA.

Leckie, C. & Kotagiri, R. (2002). *"A Probabilistic Approach to Network Scan Detection*". In Proceeding of the 8th IEEE Network Operations and Management Symposium (NOMS 2002), April 15-19, 2002, Italy.

Luis Mg (2007), *"TCPDUMP*", Retrieved Jun 5, 2007, from http://www.tcpdump.org

Mellia, M., Meo, M. & Muscoriello, L. (2006). *"TCP Anomalies: Identification and Analysis. In Proceeding of Distributed Cooperative Laboratories: Networking, Instrument and Measurement*". SpringerLink, 2 July 2006.

Montgomery, C.G & George, C.R. (2007). *"Applied Statistics And Probability for Engineer, 4th Edition*". Asia: John Wiley and Sons, Inc.

MyCert. (2008), Retrieve on 6 August 2008, from http://www.mycert.org.my.

Nemeth, E., Synder, G. & Hein, T.R. (2002). *"Linux Administration Handbook*". USA. Prentice Hall PTR

Onut IV and Ghorbani AA. (2006). *"Toward a Feature Classification Scheme for Network Intrusion Detection"*, In Proceeding of the 4th annual Communication Network and Services Research Conference, IEEE, May 24-25 2006, Canada.

Phyton Software Foundation (2007), *"Phyton*", Retrieved December 5, 2007, from http://www.phyton.org

Robertson S., Siegel EV., Miller M. & Stolfo SJ. (2003), *"Surveillance Detection in High Bandwidth Environment"*. In Proceeding of IEEE Conference on the DARPA information Survivability and Exposition, IEEE, Vol. 1, 2003, pp. 130-138.

S. Chebrolu, A. Abraham, J. P. Thomas , (2004) *"Feature Deduction and Ensemble Design of Intrusion Detection System"* , Computer and Security. Elsevier Ltd. Vol 24, 13 November 2004, pp. 295-307.

Shahrin Sahib@Sahibuddin, Mohd Faizal Abdollah, Robiah Yusof, Siti Rahayu Selamat (2007a) *"Real Time Traffic Classification For Intrusion Detection System"* , Proceeding REACH07, Kuala Lumpur.

Shahrin S., Othman M, Robiah Y, Faizal MA, Rahayu, S.S, Nazrulazhar, B., Aslinda, H, Fairuz, (2007b), *"MI. Technical Report on Network Health Check"* .

Sourcefire Inc.(2008) , *"Snort"*, Retrieve on 2 January 2008, from http://www.snort.org

V. Paxson (1999), *"Bro: A System for Detecting Network Intruders in Real-Time"*, Proc. 7th USENIX Security Symposium, Jan. 1998

W. Lee and S. J. Stolfo, (2004) *" A Framework for Constructing Feature and Model for Intrusion Detection Systems"*, In Proceeding ACM Transactions on Information and System Security, Vol 3, No. 4, November 2004, 227-261.

Wang, L., Zhao, X., Pei, D., Bush, R., Massey, D. & Zhang, L. (2003). *"Protecting BGP Routes to Top-Level DNS Servers" ,*In Proceeding of IEEE Transaction on Parallel and Distributed Systems, Vol.14, No.9, 322-331, USA.

Ye, N & Chen, Q. (2001). *"An Anomaly Detection Technique Based on Chi-Square Statistic for Detecting Intrusion into Information System"*, International Journal of Quality and Reliability Engineering, Vol, 17, pp. 105-112.

Ye, N., Emran M.S., Li, X. & Chen, Q. (2001). *"Statistical Process Control for Computer Intrusion Detection"*. In Proceeding DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01, June 12-14, 2001, USA.