Network-based Intrusion Detection Model for Detecting TCP SYN flooding

Urupoj Kanlayasiri
Department of Computer Engineering
Kasetsart University
Bangkok, Thailand
E-Mail: g4265106@ku.ac.th

Surasak Sanguanpong
Department of Computer Engineering
Kasetsart University
Bangkok, Thailand
E-Mail: nguan@ku.ac.th

Abstract: This paper presents a method for detecting TCP SYN flooding attack using BENEF model. Our model relies on the significant parameters of anomalous network packets, the statistic of system behavior, and the decision with threshold and fuzzy rule-based technique. With fuzzy technique, rules or a set of rules corresponding with the appropriate membership value are designed for analysis and to find the final decision. Our first prototype employs BENEF model to implement the network-based intrusion detection system. Current implementation is experiment with TCP SYN flooding attacks.
Key words: Network-based Intrusion Detection, network intrusion, TCP SYN flooding attack

1. Introduction

Network security is a seriously concerned topic for both private and public communications. Many efforts for protecting network system and handling intrusive action are proposed, for instance, Firewall, Crypto-based System, and Intrusion Detection System (IDS). Firewall is software or hardware system designed to filter out unwanted messages and allow legal communications. Crypto-based system uses cryptography technology to prevent the confidential data and perform the authentication. IDS is an automated system intended to detect computer intrusions. The main goal of IDS is to identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrations [1]. However, there is no best solution among the above methods that can provide the detection, prevention, and counter-attack of all intrusive patterns. To accomplish the security goal, it is very important to select the appropriate technique adapting to the organization. Frequently, the blending of each method is preferable.

It is essential to have a tool for detecting the computer intrusion. IDS also performs at network and host level for detecting various attacks. There are two domains of intrusion detecting techniques based on the detection method: misuse detection and anomaly detection. Misuse or knowledge-based is an attempt to recognize the well-known flaws or vulnerabilities of software or computer system. It can detect the general attack signatures that stem from the known holes such as exploiting a software bug. Anomaly or behavior-based detection, on the other hand, can be identified intrusions by unusual behavior of operations. According to Kumar's paper [2], "Anomaly detection attempts to quantify the usual or acceptable behavior and flags other irregular behavior as potentially intrusive"

Traditionally, the audit source location distinguishes among IDSs based on the kind of input information they analyzed. There are two categories such as host-based and network-based intrusion detection system. The host-based IDS monitors a single host machine employing the audit trails of a host operating system as a main source of input. It was regarded as a forerunner of the network-based intrusion detection system. The host-based IDSs, which have been widely developed in the past several years, can detect both anomaly and misuse behaviors. Generally, they often appear as the system embedded in a risky machine. The network-based IDS monitors any numbers of hosts in network segment. It peruses the audit trails of multiple hosts and network traffic to identify the intrusion signatures. This novel approach is a stand-alone system that can detect an intruder invaded into any systems via a computer network. Unlike the host-based IDS, it does not depend upon any operating system.

However, it is very difficult, perhaps impossible in some cases, to build an IDS that can completely detect all kinds of intrusions. Although many approaches were presented, there is no one best solution or technique for constructing the perfect system. The system may lead either "false-positive" or "false-negative" errors because of uncertain decisions. False-positive error is the mistake of the system that appears when IDS classifies an action as anomalous or a possible intrusion when it is a legitimate action. A false-negative error occurs when an actual intrusive action is allowed to pass as non-intrusive behavior.

The rest of this paper is organized as follows. In Section 2, the related works and relevant researches are provided. We describe the characteristic and behavior of Denial of Service (DoS) attack class in Section 3. In Section 4, we propose the concept of our model, system architecture, primarily verification, and formal framework to detect TCP SYN flooding attack. Finally, Section 5 gives conclusions and future works.

2. Related works

An intrusion detection problem has been widely studied in the computer security field. There are many models designed to identify the computer intrusions. Different model architectures are formed by different approaches. Nowadays, in the realm of intrusion detection, several ways were brought to develop the efficient system. Recently, a vast number of detection approaches are proposed [3]. Several theories were applied to contribute a more powerful detection but there is no best method that covers all class of penetrations. Each approach may be technically appropriate to identify a specific scenario of security violations. Some techniques may use to run cooperatively together with others to yield a better detection.

The simple method to identify the computer intrusion is Threshold detection [4]. This rudimentary form utilizes a reasonable threshold value to give an alarm when the number of occurrences of suspicious event surpasses it. The most difficult in implement using this approach is how to select the suitable measured threshold for each specific event. Threshold analysis is suitable for detecting obvious intrusive patterns. In another point of view, ability in intrusive detection of uncommon event is poor. Hence, this approach is often used as a sub-component operation to enhance the efficiency of a large IDS. The Network Anomaly Detection and Intrusion Reporter or NADIR [5] is an example of this threshold detection.

To distinguish an abnormal event from a normal activity, anomaly detection has widely used in recent years. Auditing trails are employed to define the normal pattern. System and/or user profile are established the normal usage over a duration of time. Typically, the two primary types of anomaly detection are statistical profile-based and rule-based [6]. Profile-based anomaly detection utilizes statistical method to identify the behavior but the rule-based detection utilizes sets of rules. The well-known profile-based anomaly detection system is the Intrusion Detection Expert System (IDES) [7]. The Wisdom and Sense (W&S) [8] is the example of rule-based anomaly detection system.

Misuse detection is a suitable approach to detect the exactly known vulnerabilities. The misuse detection appears as a sub-component in most intrusion detection systems. Intrusion signatures are usually specified as a sequence of conditions and events that lead to a break-in. Most systems using this approach deal with the adversary by employing the pattern-matching technique [9]. The rule-based detection system decides the event as a penetration when audit records are parsed the predefined rules. A rule or a sequence of rules describes the suspicious event that becomes an attack as an IF-THEN expression.

The model-based detection [10] represents the dangerous scenario as a high-level abstraction, unlike an audit record. The main goal is to build the model identified the behavior of intrusion. Model-based technique differs from current rule-based technique, which simply attempts to match the pattern with audit records.

3. Denial of Service attack

Denial of Service (DoS) is a common attack that has been used for a long time. DoS attack is a method intended to exhaust the network and station resources. There are many ways to compromise computer systems, for instance, Smurf [11], Teardrop, and TCP SYN flooding [12].

Smurf or ICMP Denial of Service attack is very old technique. It does not destroy any system components except to consume network bandwidth. The attacker composed a vast number of packets with spoofed source address. These packets are ICMP echo request types with source address of victim address sending to many innocent hosts. This means that all receivers will response back the victim with a large number of ICMP echo replies. Consequently, the bandwidth of victim's network is wastefully utilized. However, today, modern routers or network devices have a functional capability to protect it.

Teardrop attack makes target's operating system confused. The packets with unacceptably wrong fragmentation are generated and sent to a victim. And then, the memory-copy operation of a vulnerable target system is failure.

TCP SYN flooding is one of the most common of DoS attack class. By taking the vulnerability of three-way handshake in TCP/IP operation, the attacker makes a lot of connection requests with spoofed address to a server. These requests are the TCP SYN packet with an unreachable source IP address. In this situation, the SYN/ACK packet is sent back to an unreachable host and there is no acknowledgement message from client to server.

TCP connection buffers of server are allocated and rapidly exhaust. Hence, new legitimate connection can not be established. The process of this operation is given as follows in Figure 1.

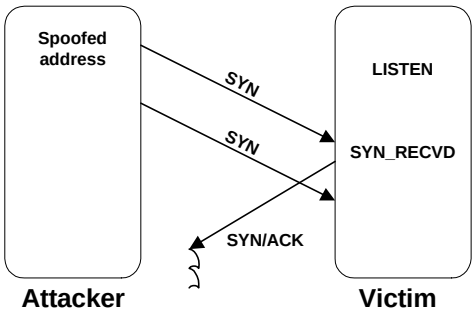

Figure 1: TCP SYN flooding attack

We give a brief summary and the characteristic of DoS attack describing in term of protocol level, attack patterns, and affected results in Table 1.

Table 1: Characteristics of some well-known techniques in Denial of Service attack class

Method	Level	Pattern	Result
Smurf	ICMP	A lot of echo ICMP request packets	Consume Network bandwidth
Teardrop	IP	Unacceptable fragmentation	System confuses
TCP SYN flooding	TCP	SYN requests with fake source	Service not respond

4. BENEF Model

In our approach, a detection model for network-based intrusion detection system is proposed. The key idea is to scrutinize an extensive set of features that were extracted from network packets. Additionally, the system employs network configuration, environment information, and system behavior to aid the decision. We use threshold detection and rule-based fuzzy-logic [13] technique to conclude final decision.

The BENEF (Behavior Statistic, Network Information Base, and Fuzzy-logic Decision) model can be categorized into both anomaly and misuse detection. This model is intended to detect the DoS attack. In this paper, we emphasize only the TCP SYN flooding as a case study.

4.1 Basic concept

Traditionally, the host-based intrusion detection system [14] is usually embedded into local host. It employs audit trails as a main source of input and can only detect the computer intrusion that breaks through local station. In this research, we propose a detecting model for network-based intrusion detection. This model performs detection methods on bus or share based networks. The main source of input is network packets and other environment information. It utilizes not only threshold detection technique to complete decision but also a fuzzy-logic to give the final answer.

4.2 System architecture

The system architecture mainly comprises 3 components: Feature Selector (FS), Pre-Detector (PD), and Decision Engine (DE). The overall is shown in Figure 2.

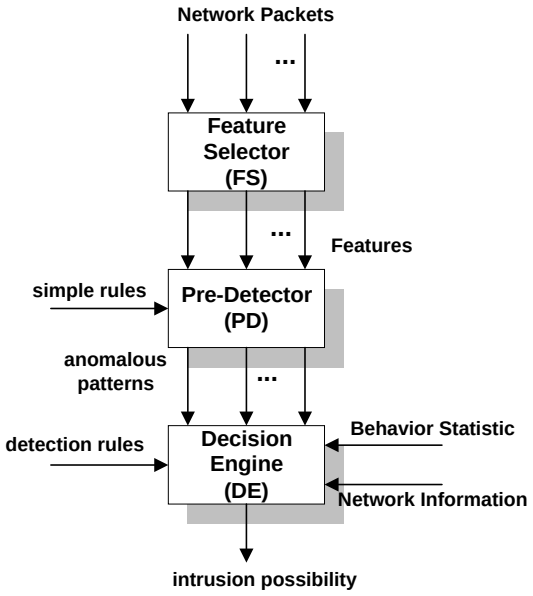

Figure 2: System architecture

4.2.1 Feature Selector

The main task of FS is to capture all packets in Ethernet local network. Then, packets are categorized by station. The FS will extract the important parameters (features) and other information from network packets. These selected features are used in Pre-Detector component. The structure of FS is shown in Figure 3.

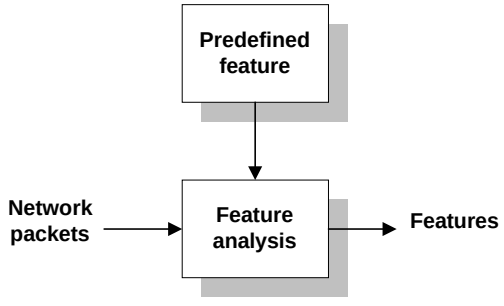

Figure 3: Feature Selector component

The features, in our case study, TCP SYN flooding attack, are such as destination IP address, destination port number, TCP flags, window size, the interval value of two adjacent sequence numbers, and interval time of two-consecutive packets. These features have been prior defined in the Predefines Feature component.

The Predefined Feature (K) is in the form of 6-tuples of parameters. Its characteristic can be defined system as follows.

K = (SYN, DA, DP, W, ΔSEQ, ΔT)

Each parameter has the following meaning:

SYN is a flag on TCP header. It identifies a connection request in three-way handshake.
DA is a destination IP address of packet.
DP is the destination port number.
W is a window size of TCP segment.
ΔSEQ is an interval value of two sequence numbers.
ΔT is an interval time of two consecutive TCP segment.

4.2.2 Pre-Detector

Pre-Detector is a preliminary analysis part of detecting phase. It analyzes events with a set of detection rule. To identify intrusive patterns, by considering the first 4 features of K, sequence of packets must conform to several conditions. First, a packet must have the SYN flag in its TCP header. Secondly, each packet has the same destination IP address (DA) and the same destination port (DP). Finally, the packet must have the same window size (W). These conditions are described with the following rule:

```
START: IF flag is SYN THEN
          IF DA are same THEN
             IF DP are same THEN
                IF W are same THEN
                      Goto Decision Engine
                   ELSE goto START
                ELSE goto START
             ELSE goto START
          ELSE goto START
```

After the anomalous events pass under conditions defined by the rules, the output events will be fed to Decision Engine to calculate the possibility of intrusion.

4.2.3 Network Information Base

To achieve a more powerful detection, network information is also considered. In our case study, we maintain two information bases (1) IP-MAC address table and (2) IP-PORT service table. An IP-MAC address table contains *N*x*M* matrix where *N* and *M* are the number of hosts and their MAC addresses respectively. An IP-PORT service table describes the services of each station. Two tables are shown as Figure 4.

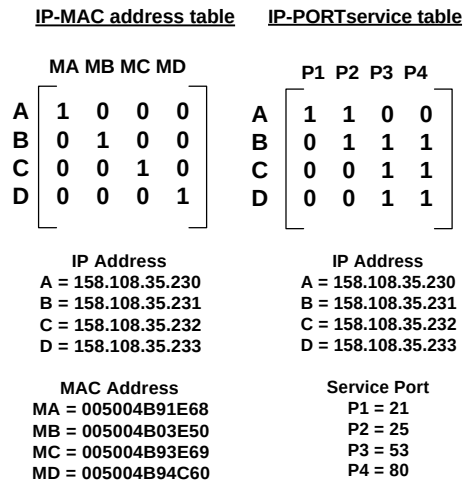

Figure 4: Network Information Bases

For some intrusive situations, it can be clearly detected by inspecting network information. For example, we use IP-MAC address table to detect insider intrusion in the case of both adversary and victim locating on the same network. It is very easy to detect the insider attacker generating SYN packets that have a couple of outsider IP address (spoofed IP address) and insider MAC address (except gateway address).

4.2.4 System Behavior Statistics

To identify the intrusion pattern, the behavior of normal event compare to anomalous pattern may be considered for better detection. For example, we may collect various normal behavior statistics based on IP-PORT service table. The example of http service in our experimental server is shown in Figure 5.

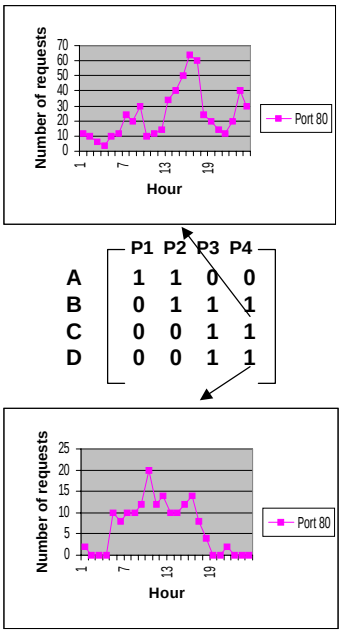

Figure 5: System Behavior Statistics

The statistic of network behavior cooperating with network information and simple rules (as described in Section 4.2.2 and 4.2.3) could be utilized as well. However, it is not included in our first prototype.

4.2.5 Decision Engine

Decision engine receives Network Information Base and System Behavior Statistic and then employs fuzzy-logic principle to decide what pattern is an intrusion. The output is in the form of percentage of intrusion possibility. In our case study, there are two significant features, ΔSEQ and ΔT. ΔSEQ is an interval value of sequence numbers of two TCP segments and ΔT is the interval time of two consecutive TCP segment. These two values could not be exactly specified with any well–form formula. Thus, the fuzzy-logic technique is employed. The fuzzy rule-based system starts with the fuzzification. We assign membership values to each feature. The membership values are derived from experiments. Membership function defines values into 3 levels, namely, low (L), medium (M), and high (H) as shown in Figure 6 and Figure 7.

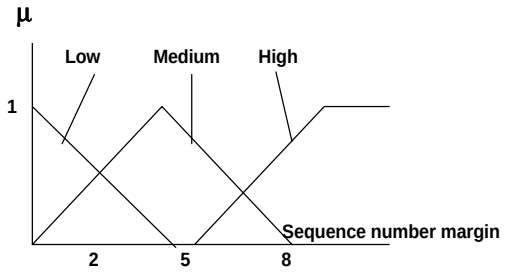

Figure 6: Membership values of ΔSEQ

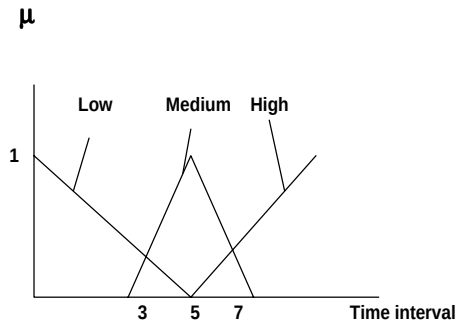

Figure 7: Membership values of ΔT

Additionally, the variable R is the result of detection. We create a set of rules in an IF-THEN form. These rules are derived from our experiments in detection of TCP SYN flooding attack and the surveying of many hacking reports [15]. The rules are described as follows.

Rule1: IF ΔSEQ=L AND ΔT=L THEN R=H
Rule2: IF ΔSEQ=L AND ΔT=M THEN R=M
Rule3: IF ΔSEQ=L AND ΔT=H THEN R=M
Rule4: IF ΔSEQ=M AND ΔT=L THEN R=H
Rule5: IF ΔSEQ=M AND ΔT=M THEN R=L
Rule6: IF ΔSEQ=M AND ΔT=H THEN R=M
Rule7: IF ΔSEQ=H AND ΔT=L THEN R=M
Rule8: IF ΔSEQ=H AND ΔT=M THEN R=M
Rule9: IF ΔSEQ=H AND ΔT=H THEN R=L

In our first prototype, a simple TCP SYN flooding attack is simulated, we measure the average of ΔSEQ and ΔT. The values are 3 and 6 respectively. Following the defuzzification and interpretation of membership function from Figure 6 and 7, ΔSEQ and ΔT give the membership value in each region as shown in Table 2.

Table 2: The result of testing experiment.

Value	Low	Medium	High
ΔSEQ = 3	0.5	0.8	-
ΔT = 6	-	0.5	0.3

From Table 2 these values match the Rule 2, 3, 5, and 6. We can compute the result from each rule as the following:

Rule2:
IF ΔSEQ = L (0.5) AND ΔT = M (0.5)
THEN R = M (0.5)

Rule3:
IF ΔSEQ = L (0.5) AND ΔT = H (0.3)
THEN R = M (0.3)

Rule5:
IF ΔSEQ = M (0.8) AND ΔT = M (0.5)
THEN R = L (0.5)

Rule6:
IF ΔSEQ = M (0.8) AND ΔT = H (0.3)
THEN R = M (0.3)

Finally, from the result of each rule above, we construct the graph and specify the possibility of intrusion by calculating the Centroid of graph as displayed in Figure 8.

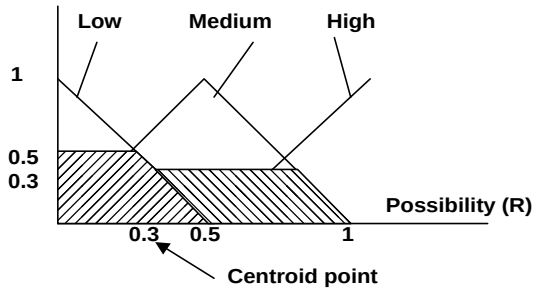

Figure 8: The possibility of intrusion

In our experiment, it can be deduced that the possibility of TCP SYN flooding occurrence in this situation is about 0.3 or 30%.

5. Conclusion and future works

We propose a powerful network-based intrusion detection model for detecting TCP SYN flooding attack. This paper describes the progression and formal framework of our intrusion detection system using this model. Now the prototype followed BENEF model is developing and testing concurrently.

Our future work will focus on a development of the system cooperated with a completed operation of behavior statistic component and refinement of fuzzy rule sets for a precise decision. Furthermore, the proper network information base and adequate statistics of system behavior will be adapted for better detection.

6. Acknowledgement

We would like to thank very much Pakorn Waewsawangwongse for providing us the good idea design of rule-based fuzzy decision technique. Thank for the helping of Chalermkon Chongsanguan and Naruchit Chinawong for contributing us some testing data as well as developing the first prototype and evaluating our approach.

7. References

[1] Biswanath Mukherjee, L Todd, Heberlein, and Karl N. Levitt. Network Intrusion Detection, IEEE Network. 8(3):26-41, May/June 1994.

[2] Kumar S. Classification and Detection of Computer Intrusions, Ph.D. Thesis, Department of Computer Sciences, Purdue University, W.Lafayette, IN 1995.

[3] Debar H, Dacier M, and Wespi A. Towards a Taxonomy of Intrusion detection Systems, Computer Network 31:805-822, 1999.

[4] Sebring M, Shellhouse E, Hanna E, and Whitehurst R. Expert System in Intrusion Detection: A Case Study, Proceedings of the 11th National Computer Security Conference: 85-91, October 1988.

[5] Jackson K, DuBois D, and Stalling C.An Expert System Application for Network Intrusion Detection, Proceeding of the 14th National Computer Security Conference: 215-225, October 1991.

[6] Koral I, Richard A, Kemmerer, and Phillip A. State Transition Analysis: A Rule-Based Intrusion Detection Approach, IEEE transactions on software engineering. 21(3):181-199, Mar 1995.

[7] Lunt T, Tamaru A, Gilham F, Jagannathan R, Jalai C, Javitz H., Valdes A, and Neumann P. A Real-Time Intrusion Detection Expert System, SRI CSL Technical Report, SRI-CSL-90-05, June 1990.

[8]Vaccaro H and Liepins G. Detection of Anomalous Computer Session Activity, Proceeding of the IEEE Symposium on Research in Security and Privacy: 280-189, May 1989.

[9] Kumar S and Eugene S. An Application of Pattern Matching in Intrusion Detection, Technical Report CSD-TR-94-013 Purdue University, IN June, 17 1994.

[10] Garvey T and Lunt T. Model-based Intrusion Detection, Proceedings of the 14th National Computer Security Conference: 372-385, October 1991.

[11] Stephen N. Network Intrusion Detection An Analyst's Handbook, NewRiders, 1999.

[12] Chuba C, Krsul I, Khun M, Spafford E, Sundram A, and Zamboni D.Analysis of Denial of Service Attack on TCP, IEEE Symposium on Security and Privacy, 1997.

[13] Timothy J. Fuzzy Logic With Engineering Applications, International edition, McGRAW-HILL. 1997.

[14] Kanlayasiri U, Sanguanpong S, and Jaratmanachot W. A Rule-based Approach for Port Scanning Detection, Electrical Engineering Conference Thailand, 2000.

[15] deamon9. Project Neptune, Phrack Magazine, 7(48), 1996.