

國立交通大學

電信工程學系碩士班

碩士論文

針對埠或位址掃描做快速偵測之
適應性接續假設測試



Adaptive Sequential Hypothesis Testing for Fast
Detection of Port/Address Scan

研究生：林建成

指導教授：李程輝 教授

中華民國九十六年七月

針對埠或位址掃描做快速偵測之適應性接續假設測試

Adaptive Sequential Hypothesis Testing for Fast Detection of
Port/Address Scan

研究生：林建成

Student: Jian-Cheng Lin

指導教授：李程輝 教授

Advisor: Prof. Tsern-Huei Lee

國立交通大學

電信工程學系碩士班

碩士論文



A Thesis

Submitted to Institute of Communication Engineering
Collage of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of Requirements

for the Degree of

Master of Science

in

Communication Engineering

July 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

針對埠或位址掃描做快速偵測之 適應性接續假設測試

學生：林建成

指導教授：李程輝 教授

國立交通大學

電信工程學系碩士班

中文摘要

隨著網路應用服務的增加，網路安全的議題也越來越受到廣泛的重視。其中埠或位址掃描這種異常的行為，是網路入侵的一個重要途徑。早期偵測這些埠或位址掃描的技術，是建立於惡意行為的主機具有較高掃描率的基礎上。但是這種方式對於偵測某些慢速的掃描並不適用，而且攻擊者一旦獲知發出警戒的門檻值，便能輕易的躲過這種偵測。為了解決這個問題，接續假設性測試便成為偵測這種掃描的另一種替代方案。這種方式可以藉由第一次連線要求的成功率之不同，來判斷發送者為正常或具有惡意攻擊行為的主機。但是假如無法知道正常與異常主機不同的連線成功率為何，其誤判的機率便會遠高於理想值。在這篇碩士論文中，我們比較了幾種以接續假設性測試為架構的技術，並且發現在實際未知連線成功率的網路中，這些基本的接續假設性測試並不適用。因此，我們提出在此測試法的基礎架構上，加入了一個簡單的適應性演算法，可以準確的估計出這些機率值。而從模擬的結果也顯示出，這個適應性的估計演算法對於原本的接續假設性測試法有極大的改善，因為它使原本對於埠或位址掃描的測試法更加健全與完備。

Adaptive Sequential Hypothesis Testing for Fast Detection of Port/Address Scan

Student: Jian-Cheng Lin

Advisor: Prof. Tsern-Huei Lee

Institute of Communication Engineering

National Chiao Tung University

Abstract

As more and more network applications and services are provided, the topic of network security becomes more and more important. The behavior anomaly of port/address scans is a way to intrude hosts on the Internet. Early detection techniques of port/address scans are based on the observation that malicious hosts could send scans with high scanning rates. But such approaches are not suitable to detect scanners with lower scanning rate. Once the threshold of scanning rate for generating alerts is known to the attackers, the detection will be easily evaded. In order to overcome the problems, sequential hypothesis testing is an alternative detection technique. According to the probabilities of success for the first-contact connection attempts sent by the hosts, sequential hypothesis testing can detect the senders as benign or malicious. If these probabilities are unknown, the false positive and false negative rates could be much larger than the desired values. In this thesis, we compare several techniques based on sequential hypothesis testing and realize these techniques inadequate for a real network. Therefore, we propose a simple adaptive algorithm which provides accurate estimation of these probabilities. Simulation results show that the proposed adaptive estimation algorithm provides a great improvement for sequential hypothesis testing.

致謝

首先，要感謝我的指導教授—李程輝老師，在我研究所的求學過程中悉心地指導與教誨，並且適時的給予實用的建議與鼓勵。從老師的身上，我看到了一位真正的學者對於研究的認真態度與無比熱忱，這是讓我由衷地欽佩和感動的。在老師的指導下，讓我初窺到學術殿堂的門徑，也讓我學習到做研究應有的態度與方法，實在是獲益良多。

感謝 NTL 實驗室的所有夥伴們，這兩年來的朝夕相處，讓我感受到無比的溫暖。謝謝景融、郁文、迺倫、以及瑋哥，眾位學長姐的關心與照顧讓我感念在心；謝謝嘉旂、柏庚、和登煌，這些日子一起修課一起通宵熬夜的革命情感令我難以忘懷；謝謝北極、耀誼、世弘、明鑫、凱文、西西搭，你們這群可愛的學弟們，讓我的碩士生涯過得很愉快，也充滿了各種美好的回憶。

最後，更要感謝我的父母與其他親人好友們，謝謝你們永遠給我無限的支持與鼓勵，讓我有信心有勇氣去面對各種挑戰。因為有你們，是讓我不斷向前邁進的原動力！

謹將此論文獻給所有愛我與我愛的人

2007 年 7 月 新竹交大

Contents

中文摘要	i
English Abstract	ii
誌謝	iii
Contents	iv
List of Tables	vi
List of Figures	vii
Chapter 1 Introduction	1
Chapter 2 Background	5
2.1 Scanning Worms	5
2.2 Scan Detection and Suppression	7
2.3 False Alarm	9
2.3.1 False Positive & False Negative	9
2.3.2 Probabilities	10
Chapter 3 Related Works	11
3.1 First-Contact Connection Requests	11
3.2 Sequential Hypothesis Testing	13
3.2.1 Model	13
3.2.2 Upper and Lower Thresholds	16
3.2.3 Log of Likelihood Ratio	17

3.2.4	Number of Observations to Select Hypothesis	18
3.3	Simplified Sequential Hypothesis Testing	19
3.3.1	Modification from SHT	20
3.3.2	Hardware Implementation	20
3.3.3	Algorithm	22
3.4	Reverse Sequential Hypothesis Testing	23
3.4.1	Model	24
3.4.2	Proof of Optimized Algorithm	25
3.4.3	Log of Likelihood Ratio	27
Chapter 4	Adaptive Sequential Hypothesis Testing	29
4.1	Scheme 1	29
4.2	Scheme 2	33
4.3	Implementation	34
Chapter 5	Simulation Results	38
5.1	SHT with known θ_0 and θ_1	38
5.2	SHT with unknown θ_0 and θ_1	45
5.3	Adaptive Sequential Hypothesis Testing	48
Chapter 6	Conclusion	54
Bibliography	55

List of Tables

Table 2.1	Definition of false positive and false negative.....	9
Table 2.2	Example of false positive and false negative	10
Table 5.1	The step sizes of failure and success for SHT.....	41
Table 5.2	SHT, θ_0 and θ_1 are known	42
Table 5.3	Simplified SHT, θ_0 and θ_1 are known	43
Table 5.4	RSHT, θ_0 and θ_1 are known	44
Table 5.5	SHT, θ_0 and θ_1 unknown, guess 0.8 and 0.2	46
Table 5.6	RSHT, θ_0 and θ_1 unknown, guess 0.8 and 0.2.....	47
Table 5.8	Adaptive SHT, $N_G \geq 0$ & $N_B \geq 0$ (0%).....	49
Table 5.9	Adaptive SHT, $N_G \geq 200$ & $N_B \geq 50$ (25%)	50
Table 5.10	Adaptive SHT, $320 \leq N_G \leq 480$ & $80 \leq N_B \leq 120$ (40~60%)	51
Table 5.11	Adaptive SHT, $N_G : N_B$ (40~60%)	52
Table 5.12	θ_0 and θ_1 of Adaptive SHT, $N_G : N_B$ (40~60%)	53

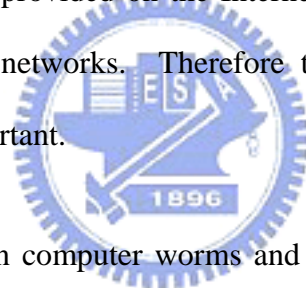
List of Figures

Figure 2.1	Spreading and propagation of scanning worms.....	6
Figure 2.2	Preventing “inside” from “outside”	8
Figure 3.1	X_i , the outcomes of FCC requests from r to l_i	12
Figure 3.2	Flow diagram of sequential hypothesis testing	15
Figure 3.3	A log scale graph of $\Lambda(\mathbf{X}_n)$ for SHT	16
Figure 3.4	Connection cache.....	21
Figure 3.5	Address cache.....	22
Figure 3.6	Algorithm for the simplified SHT	22
Figure 3.7	A log scale graph tracing the value of $\Lambda(\mathbf{X}_n)$ for SHT.....	24
Figure 3.8	A log scale graph tracing the value of $\bar{\Lambda}(\mathbf{X}_n)$ for RSHT.....	24
Figure 4.1	Adaptive procedure I.....	32
Figure 4.2	Adaptive procedure II.....	34
Figure 4.3	List of connection	35
Figure 4.4	Connection Table.....	35
Figure 4.5	Address Table	36
Figure 4.6	The modified algorithm for SHT.....	37

Chapter 1

Introduction

As the computer and network technologies advance rapidly, more and more services and applications are provided on the Internet. Today, many people can't live without computers and networks. Therefore the topic of network security becomes more and more important.



As time goes by, modern computer worms and viruses can spread at a speed much faster than human intervention. A computer worm automatically spreads from computer to computer by exploiting a software vulnerability that allows an arbitrary program to be executed without proper authorization. In recent years, people discovered many kinds of worms, such as the Code Red [11], Nimda [12], and Slammer [6], which infected thousands upon thousands of computers on the Internet in a short period of time and caused great damage to our society. It's important to prevent the majority of vulnerable systems from being detected and minimize the damage caused by computer worms. Fast and accurate detection of worms when they are spreading is, therefore, helpful to solve the problems.

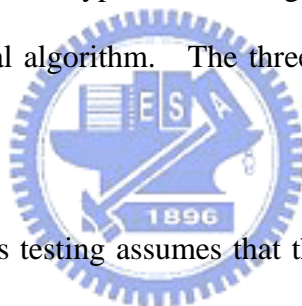
Current technologies for computer worm detection can be classified into three

categories – protocol analysis, pattern matching, and behavior anomaly. First, protocol analysis is used to inspect if there are misused protocol fields in the header of a packet. The header of a packet sent by a malicious host is usually spoofed or altered. The malicious hosts can be detected according to the misuse of fields. Then, pattern matching is used to look for specific patterns in the payload of a packet or across packets. The signatures of worms, e.g. specific unique patterns or strings of malicious codes, can be extracted and then utilized for worm detection. Although pattern matching is accurate, it is limited to detect worms with identified signatures. If the signatures of new worms are not created promptly, the majority of vulnerable systems could be infected.

Finally, behavior anomaly can be used to detect and prevent port/address scans because an infected host is likely to behave differently from a normal host. For example, an infected host could try to infect other vulnerable host on the Internet with port or address scanning. Therefore, we can detect the infected host with the observation that it has high new connection attempt rate or high failure rate of new connection attempts. Because the technique based on behavior anomaly can detect worms without signatures, it is useful to deal with new computer worms.

Seeing that most of current intrusion detection systems (IDS) based on the technique of pattern matching can't detect new and unknown malicious attacks or scans, network behavior anomaly detection (NBAD) is receiving more and more attention. Recently, more and more IDS adopted the mechanism based on behavior anomaly detection. For example, the Network Security Monitor (NSM) [13] and Snort [14] are designed according to simple observation of high scanning rate by an infected host.

In the paper [1], a technique of sequential hypothesis testing for scan detection is proposed, and the algorithm is called Threshold Random Walk. The technique is based on the observation that success rate of a connection attempt sent by a malicious host is much lower than the success rate of a connection attempt sent by a benign host. A random walk of each host is moving upward if a connection attempt is a failure, or moving downward if a connection attempt is a success. A host is detected as malicious if the position of its random walk is greater than the upper threshold or as benign if it is smaller than the lower threshold. A simplified sequential hypothesis testing [3] is suitable for both software and hardware implementations. It modified the step sizes of moving upward and downward to be identical. The reverse sequential hypothesis testing [2] can detect malicious host slightly faster than the original algorithm. The three algorithms will be review in Chapter 3.



The sequential hypothesis testing assumes that the success rates of connection attempts sent by benign and malicious hosts are known. They are used to compute the step sizes of moving upward and downward. But in fact, the success rates of connection attempts could be unknown. Therefore, we develop the sequential hypothesis testing with an adaptive procedure which can estimate the success rates of connection attempts based on their outcomes. It can provide estimates close to real values and reduce both the false positive and negative rates

The rest of this thesis is organized as follows. In Chapter 2, we introduce some background about scanning worms, scan detection and suppression, and the definition of false positives and false negatives. In Chapter 3, we review the sequential hypothesis testing, the simplified sequential hypothesis testing, and

reverse sequential hypothesis testing. In Chapter 4, we present our proposed adaptive algorithm for estimation of success rate of connection attempts. Simulation results are provided in Chapter 5. Finally, we draw conclusion in Chapter 6.



Chapter 2

Background

2.1 Scanning Worms

A computer worm is a form of malware that spreads from host to host without human intervention. A scanning worm locates vulnerable hosts by generating a list of addresses to probe and then contact them. Figure 2.1 illustrates that worms can self-propagate among the hosts exploiting security or policy flaws in widely-used services [10]. An infected host initiates scans and infects the other benign hosts. Subsequently, the benign hosts become infected ones and then join the army of scanning. Finally all the hosts on the Internet will be infected.

This addresses list may be generated sequentially or pseudo-randomly. Local addresses are often preferentially selected because the communication between neighboring hosts will likely encounter fewer defenses [5]. Scans may take the form of TCP connection requests (SYN packets) or UDP packets. In the case of the connectionless UDP protocol, it is possible for the scanning packet to also contain the body of the worm, such as the Slammer worms [6].

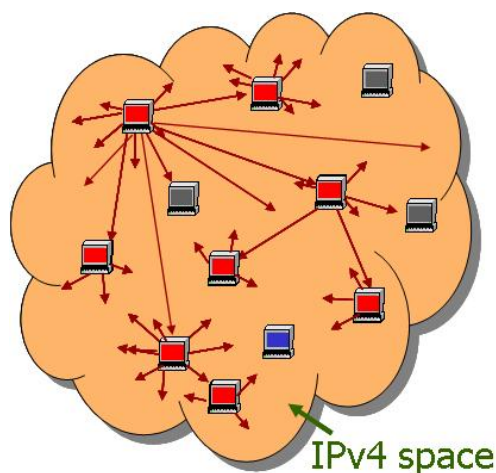


Figure 2.1 Spreading and propagation of scanning worms

Scanning worms probe attempts to determine if a service is operating at a target IP address and then discover new victims. They have two basic scanning types – horizontal scans, which look for an identical service on a large number of hosts, and vertical scans, which examine an individual host to discover all running services.

There are many kinds of techniques to generate a list of addresses for scanning worms, such as linear scanning of an IP address space (Blaster), fully random (Code Red), a bias toward local address (Code Red II and Nimda), or even more enhanced techniques (Permutation Scanning). While more and more scanning worms change their style of scanning to avoid being detected, all of them still have two common properties as follows. *Most of the scanning attempts may result in failure, and the infected hosts will send many connection attempts* [4]. As long as we look for a class of behavior rather than specific worm signatures, most new worms will be detected.

In the next chapter, we will introduce three kinds of existing on-line algorithms to detect the presence of scanning worms by observing network traffic. These algorithms based on the sequential hypothesis testing (SHT) can differentiate between

infected hosts and normal hosts according to the success rate of connection attempts.

2.2 Scan Detection and Suppression

Human reaction time is inadequate for detecting and responding to fast scanning worms, such as Slammer, which can infect the majority of vulnerable hosts on the entire IP address space in a few minutes [6, 7]. Thus, today's worm detection techniques focus on automated response to worms, such as quarantining infected machines, automatic generation and installation of patches, and reducing the rate at which worms can send connection attempts [8].

But, an automated response will be of little use if it fails to be triggered quickly after a host is infected. Infected hosts with high network bandwidth can send thousands of connection attempts per second, each of which has the potential to spread the infection. On the other hand, an automated response that triggers too easily will erroneously identify normal hosts as infected. It will interfere with the normal activity of these hosts and cause significant damage.

Many scan detection mechanisms rely on the observation that only a small part of addresses are likely to respond to a connection attempt at any given port. If a connection attempt is sent to an inactive host, it will also be failed. When a connection attempt does reach an active host, it would be rejected possibly because not all hosts will be running the targeted services. Thus, the infected hosts are likely to have a low rate of successful connection attempts, whereas benign hosts, which only send connection attempts when there is reason to believe that addresses will

respond, will have a higher success rate. So, we can make good use of the properties described above to detect scanning worms with malicious connection attempts.

Worm containment is designed to stop the spread of worms in a local area network or an enterprise by detecting infected machines and preventing them from contacting other systems. Current approaches to containment are based on detecting the scanning activity, and the key component for today's containment techniques is scan suppression which responds to detected infected hosts by blocking future scanning attempts. [4]

The goal of scan suppression is to prevent scanning attempts coming from "outside" inbound to the "inside". Here "inside" means the internal network of an enterprise or a laboratory, to be protected from the "outside" larger networks. Therefore, any scanning worms will be quickly detected and stopped because all of the malicious traffic will be seen by the detector. The illustration is shown as Figure 2.2.

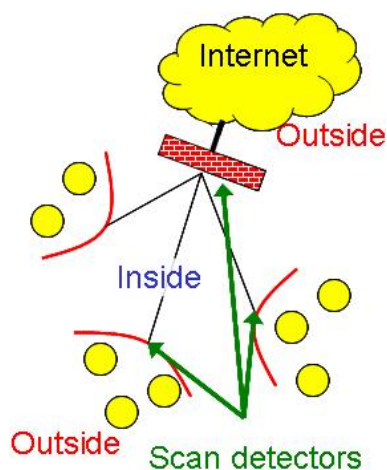


Figure 2.2 Preventing "inside" from "outside"

2.3 False Alarm

When the scan detection mechanisms determine a host is malicious or benign, it is possible to make error decisions, such as regarding as malicious when the host is benign or regarding as benign when it is infected actually. Both of them are called false alarm. We hope that the scan detection mechanism would distinguish between malicious and benign hosts as precisely as possible, and the probability of false alarm is as less as possible. So, we can use false alarm rate to judge whether an algorithm is suitable for scan detection.

2.3.1 False Positive & False Negative

The false alarm can be divided into two conditions which are false positive and false negative [9]. The former is the error of rejecting something that should have been accepted, such as finding an innocent host guilty. The latter is the error of accepting something that should have been rejected, such as finding a guilty host innocent. Table 2.1 and 2.2 will illustrate these conditions as follows.

		Actual Condition	
		Present	Absent
Test Result	Positive	Condition Present + Positive Result = True Positive	Condition Absent + Positive Result = False Positive
	Negative	Condition Present + Negative Result = False Negative	Condition Absent + Negative Result = True Negative

Table 2.1 Definition of false positive and false negative

		Actual Condition	
		Scanner	Benign
Test Result	Scanner	Actual Scanner + Result Scanner = Detection	Actual Benign + Result Scanner = False Positive
	Benign	Actual Scanner + Result Benign = False Negative	Actual Benign + Result Benign = Normal

Table 2.2 Example of false positive and false negative

2.3.2 Probabilities

The false positive rate is the proportion of negative instance that were erroneously reported as being positive. The false negative rate is the proportion of positive instance that were erroneously reported as being negative. So we can define them as follows.

$$\text{false positive rate} = \frac{\text{number of false positives}}{\text{number of negative instances}}$$

$$\text{false negative rate} = \frac{\text{number of false negatives}}{\text{number of positive instances}}$$

For scan detection, we can also define four outcomes as follow.

$$\text{false positive rate} = \frac{\text{number of scanner but actually benign}}{\text{number of total benign}} = P_{FP}$$

$$\text{false negative rate} = \frac{\text{number of benign but actually scanner}}{\text{number of total scanner}} = P_{FN}$$

$$\text{normal rate} = \frac{\text{number of benign and actually benign}}{\text{number of total benign}} = P_{NM} = 1 - P_{FP}$$

$$\text{detection rate} = \frac{\text{number of scanner and actually scanner}}{\text{number of total scanner}} = P_{DT} = 1 - P_{FN}$$

Chapter 3

Related Works

3.1 First-Contact Connection Requests

In the previous chapter, we can know that one of the main characteristics of infected hosts is that they are more likely to choose hosts that do not exist or do not have the requested service activated than benign hosts. This is because they lack precise knowledge of which hosts and ports are currently active.

Using this observation, there are several kinds of on-line algorithms to detect malicious attacks or connection attempts. The goal of these approaches is to reduce the number of observed connection attempts to flag malicious hosts, while bounding the probabilities of false positive and false negative.

An event is generated and monitored when a remote source r makes a first-contact connection (FCC) request to a local destination l . An FCC request is a connection request which is addressed to a host the sender has not previous communicated. These events are monitored because malicious scans are mostly composed of first-contact connection requests.

Only the TCP connections are considered and thus a TCP SYN packet indicates a connection request. The outcome of an FCC request is classified as either a “success” or a “failure”. It is a success if the host l replies a SYN-ACK packet or a failure if host l replies a RST packet or does not reply at all. If the request sent by r is a UDP packet, any UDP packet from l received before the timeout will be a success.

For a given remote (outside) host r , let X_i be a random variable that represents the outcome of the FCC request from r to the i^{th} distinct local (inside) host l_i , where

$$X_i = \begin{cases} 0 & \text{if the FCC request is a success} \\ 1 & \text{if the FCC request is a failure} \end{cases}$$

Figure 3.1 illustrates X_1, X_2, \dots, X_5 from r to l_1, l_2, \dots, l_5 .

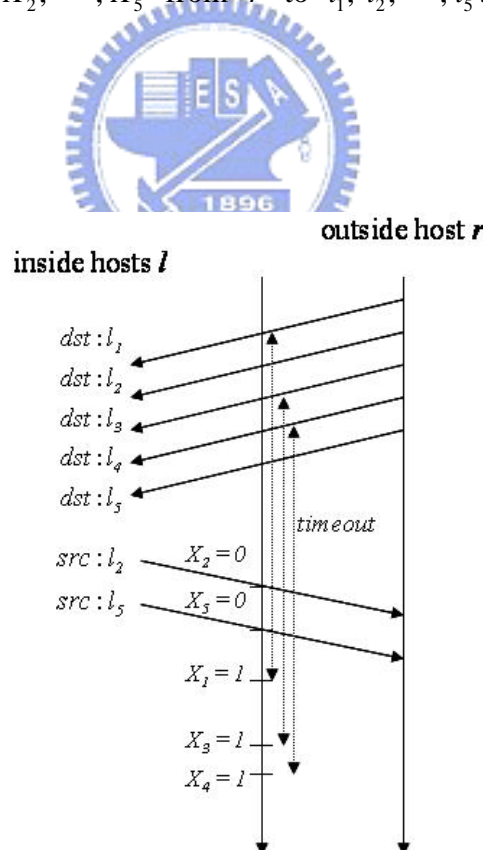


Figure 3.1 X_i , the outcomes of FCC requests from r to l_i

The outcomes X_1, X_2, \dots , are observed so that host r can be determined to be either malicious or benign. Undoubtedly, we would like to make this detection as quickly and correctly as possible. The method of *sequential hypothesis testing (SHT)* developed by Wald [1] is suitable for scanning worm detection. In the following sections, several techniques based on SHT will be introduced.

3.2 Sequential Hypothesis Testing

3.2.1 Model

In the paper [2], the technique of sequential hypothesis testing is developed. The algorithm is called *Threshold Random Walk (TRW)*. There are two hypotheses: H_0 and H_1 , where H_0 is the null hypothesis that the remote host r is benign and H_1 is the hypothesis that r is malicious.

To simplify the analysis, it is assume that, conditioning on hypothesis H_j , the random variables $X_1|H_j, X_2|H_j, \dots$ are independent and identically distributed (i.i.d) with probability mass function

$$\begin{aligned} P[X_i = 0 | H_0] &= \theta_0 & P[X_i = 1 | H_0] &= 1 - \theta_0 \\ P[X_i = 0 | H_1] &= \theta_1 & P[X_i = 1 | H_1] &= 1 - \theta_1 \end{aligned}$$

For some θ_0 and θ_1 which satisfy $\theta_0 > \theta_1$. It is because a FCC attempts is more likely to be a success from a benign host than a malicious host.

Given the two hypothesis, there are four possible decisions as follows. The decision is called a *detection* when the algorithm selects H_1 when H_1 is in fact true. On the other hand, it is called a *false negative* if the algorithm chooses H_0 . Likewise, when H_0 is in fact true, selecting H_1 constitutes a *false positive* and selecting H_0 when H_0 is called a *normal*. These four possible outcomes are represented as:

$$\begin{aligned}
 \text{Detection :} & \quad \text{P}[\text{choose } H_1 \mid H_1 \text{ is true}] = P_{DT} \\
 \text{False Negative :} & \quad \text{P}[\text{choose } H_0 \mid H_1 \text{ is true}] = P_{FN} = 1 - P_{DT} \\
 \text{False Positive :} & \quad \text{P}[\text{choose } H_1 \mid H_0 \text{ is true}] = P_{FP} \\
 \text{Normal :} & \quad \text{P}[\text{choose } H_0 \mid H_0 \text{ is true}] = P_{NM} = 1 - P_{FP}
 \end{aligned}$$

The desired performance of the TRW algorithm can be specified with the detection probability P_{DT} and the false positive probability P_{FP} . Let α represents the upper bound of false positive probability and β denote the lower bound of detection probability. In other word, we desire

$$P_{FP} \leq \alpha \text{ and } P_{DT} \geq \beta$$

where typical values might be $\alpha = 0.01$ and $\beta = 0.99$.

As the outcome of X_i is observed, we calculate the likelihood ratio:

$$\Lambda(\mathbf{X}_n) \equiv \frac{\text{P}[\mathbf{X}_n \mid H_1]}{\text{P}[\mathbf{X}_n \mid H_0]} = \prod_{i=1}^n \frac{\text{P}[X_i \mid H_1]}{\text{P}[X_i \mid H_0]}$$

where $\mathbf{X}_n \equiv (X_1, X_2, \dots, X_n)$ is the vector of outcomes observed so far.

Note that $\Lambda(\mathbf{X}_n)$ can be updated incrementally. Let $\phi(X_i)$ represent the likelihood ratio of the i^{th} observation. It holds that

$$\Lambda(\mathbf{X}_n) = \prod_{i=1}^n \phi(X_i) = \Lambda(\mathbf{X}_{n-1})\phi(X_n), \quad \Lambda(\mathbf{X}_0) = 1$$

$$\phi(X_i) \equiv \frac{P[X_i | H_1]}{P[X_i | H_0]} = \begin{cases} \frac{1-\theta_1}{1-\theta_0} > 1 & \text{if } X_i = 1 \text{ (failure)} \\ \frac{\theta_1}{\theta_0} < 1 & \text{if } X_i = 0 \text{ (success)} \end{cases}$$

The flow diagram is shown in Figure 3.2. The updated likelihood ratio $\Lambda(\mathbf{X}_n)$ is compared to an upper threshold η_1 and a lower threshold η_0 . If $\Lambda(\mathbf{X}_n) \geq \eta_1$, then the hypothesis H_0 is accepted. If $\Lambda(\mathbf{X}_n) \leq \eta_0$, then the hypothesis H_1 is accepted. More observations are needed if $\eta_0 < \Lambda(\mathbf{X}_n) < \eta_1$.

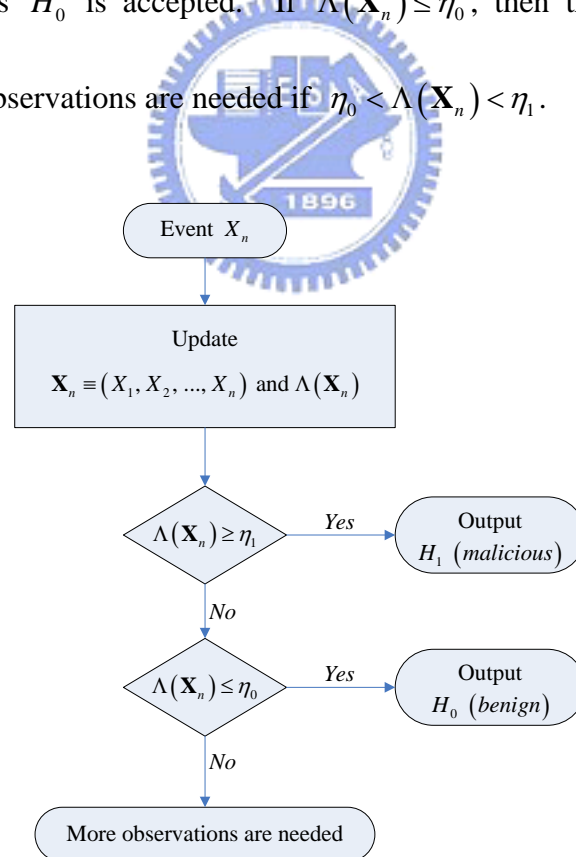


Figure 3.2 Flow diagram of sequential hypothesis testing

Figure 3.3 represents a log scale graph of $\Lambda(\mathbf{X}_n)$ when each observation X_i is added to the sequence. Each success (0) observation decreases $\Lambda(\mathbf{X}_n)$, moving it closer to the benign conclusion threshold η_0 . Each failure (1) observation increases $\Lambda(\mathbf{X}_n)$, moving it closer to the infection conclusion threshold η_1 .

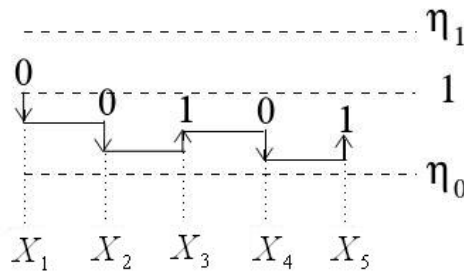


Figure 3.3 A log scale graph of $\Lambda(\mathbf{X}_n)$ for SHT

3.2.2 Upper and Lower Thresholds

To develop the algorithm described in the previous section, the thresholds η_0 and η_1 can be bounded by simple expressions of P_{FP} and P_{DT} [1].

Consider a sample path of observations X_1, X_2, \dots, X_n . The upper threshold η_1 is hit on the observation X_n and hypothesis H_1 is selected. Thus,

$$\Lambda(\mathbf{X}_n) \equiv \frac{\mathbb{P}[X_1, X_2, \dots, X_n | H_1]}{\mathbb{P}[X_1, X_2, \dots, X_n | H_0]} = \frac{P_{DT}}{P_{FP}} \geq \eta_1$$

The first probability (H_1 is selected when H_1 is true) is the detection probability

P_{DT} , and the second (H_1 is selected when H_0 is true) is the false positive probability P_{FP} .

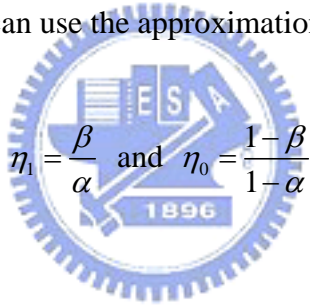
Similarly, if the lower threshold η_0 is hit and hypothesis H_0 is selected, then

$$\Lambda(\mathbf{X}_n) \equiv \frac{P[X_1, X_2, \dots, X_n | H_1]}{P[X_1, X_2, \dots, X_n | H_0]} = \frac{1 - P_{DT}}{1 - P_{FP}} \leq \eta_0$$

Therefore, the upper and lower bounds can be bounded in terms of P_{FP} and P_{DT} .

$$\eta_1 \leq \frac{P_{DT}}{P_{FP}} \quad \text{and} \quad \eta_0 \geq \frac{1 - P_{DT}}{1 - P_{FP}}$$

In real implementation, one can use the approximations $P_{FP} = \alpha$, $P_{DT} = \beta$ and set



$$\eta_1 = \frac{\beta}{\alpha} \quad \text{and} \quad \eta_0 = \frac{1 - \beta}{1 - \alpha}$$

3.2.3 Log of Likelihood Ratio

Moreover, one can use the log-likelihood ratio to simplify computation. It can be formulated as follows.

$$S_n \equiv \ln(\Lambda(\mathbf{X}_n)) = \sum_{i=1}^n Y_i = S_{n-1} + Y_n, \quad S_0 = 0$$

$$Y_i \equiv \ln\left(\frac{P[X_i | H_1]}{P[X_i | H_0]}\right) = \ln(\phi(X_i)) = \begin{cases} \ln(1 - \theta_1) - \ln(1 - \theta_0) \equiv F > 0 & \text{if } X_i = 1 \\ \ln(\theta_1) - \ln(\theta_0) \equiv S < 0 & \text{if } X_i = 0 \end{cases}$$

The log-likelihood ratio S_n is also compared to an upper threshold $\ln(\eta_1)$ and a

lower threshold $\ln(\eta_0)$. If $S_n \geq \ln(\eta_1)$, then the hypothesis H_0 is accepted. If $S_n \leq \ln(\eta_0)$, then the hypothesis H_1 is accepted. More observations are needed if $\ln(\eta_0) < S_n < \ln(\eta_1)$.

3.2.4 Number of Observations to Select Hypothesis

In this section, the average number of FCC attempts sent by a remote host to detect it as benign or malicious is calculated. The smaller the number N of observations, the faster a remote host will be identified.

For the analysis of N , the log of likelihood ratio should be used. Because S_N is the summation of N random variables Y_i where N is also a random variable, the expected value of S_N equals to the product of expected values of Y_i and N .

$$S_N = Y_1 + Y_2 + \dots + Y_N \Rightarrow E[S_N] = E[Y_i]E[N]$$

Therefore, we can derive expressions for the expected values of S_N and Y_i , conditioning on hypotheses H_0 and H_1 . The conditional expected value of N is the ratio of the conditional expected values of S_N and Y_i .

For Y_i ,

$$Y_i | H_0 = \begin{cases} \ln\left(\frac{1-\theta_1}{1-\theta_0}\right) & \text{with prob. } 1-\theta_0 \\ \ln\left(\frac{\theta_1}{\theta_0}\right) & \text{with prob. } \theta_0 \end{cases} \Rightarrow E[Y_i | H_0] = (1-\theta_0)\ln\left(\frac{1-\theta_1}{1-\theta_0}\right) + \theta_0\ln\left(\frac{\theta_1}{\theta_0}\right)$$

$$Y_i | H_1 = \begin{cases} \ln\left(\frac{1-\theta_1}{1-\theta_0}\right) & \text{with prob. } 1-\theta_1 \\ \ln\left(\frac{\theta_1}{\theta_0}\right) & \text{with prob. } \theta_1 \end{cases} \Rightarrow E[Y_i | H_1] = (1-\theta_1)\ln\left(\frac{1-\theta_1}{1-\theta_0}\right) + \theta_1\ln\left(\frac{\theta_1}{\theta_0}\right)$$

For S_N ,

$$S_N | H_0 = \begin{cases} \ln(\eta_1) & \text{with prob. } \alpha \\ \ln(\eta_0) & \text{with prob. } 1-\alpha \end{cases} \Rightarrow E[S_N | H_0] = \alpha \ln(\eta_1) + (1-\alpha) \ln(\eta_0)$$

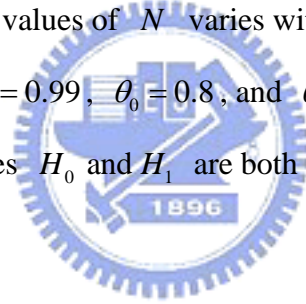
$$S_N | H_1 = \begin{cases} \ln(\eta_1) & \text{with prob. } \beta \\ \ln(\eta_0) & \text{with prob. } 1-\beta \end{cases} \Rightarrow E[S_N | H_1] = \beta \ln(\eta_1) + (1-\beta) \ln(\eta_0)$$

So, the conditional expected values of N : $E[N] = E[S_N] / E[Y_i]$

$$E[N | H_0] = \frac{\alpha \ln(\eta_1) + (1-\alpha) \ln(\eta_0)}{(1-\theta_0) \ln\left(\frac{1-\theta_1}{1-\theta_0}\right) + \theta_0 \ln\left(\frac{\theta_1}{\theta_0}\right)}$$

$$E[N | H_1] = \frac{\beta \ln(\eta_1) + (1-\beta) \ln(\eta_0)}{(1-\theta_1) \ln\left(\frac{1-\theta_1}{1-\theta_0}\right) + \theta_1 \ln\left(\frac{\theta_1}{\theta_0}\right)}$$

It represents that the expected values of N varies with the parameters α , β , θ_0 , and θ_1 . With $\alpha = 0.01$, $\beta = 0.99$, $\theta_0 = 0.8$, and $\theta_1 = 0.2$, the expected values of N conditioning on hypotheses H_0 and H_1 are both 5.41.



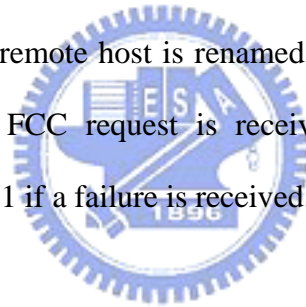
3.3 Simplified Sequential Hypothesis Testing

The huge complexity of monitoring FCC attempts of all remote hosts makes the TRW algorithm infeasible. In the paper [4], a simplified version of sequential hypothesis testing sets both the step sizes of moving upward and downward to one for the detection algorithm, and uses one bit to indicate whether or not host r has sent any connection to host l and another bit for the opposite direction. Each connection is recorded and indexed by hash the local IP address, remote IP address, and local port number for TCP protocol. A hash function is adopted to index the connections and reduce the space requirement.

3.3.1 Modifications from SHT

Using sequential hypothesis testing, each remote host has a likelihood value that a series of FCC requests from the given host reflect benign or malicious, based on how far the random walk deviates above or below the origin. The likelihood values of remote hosts are updated continuously when a FCC request is determined as a success or a failure. A successful connection request drives a random walk downward, whereas a failed connection request drives it upward.

The step sizes of moving upward and downward are both simplified to one. The likelihood value of each remote host is renamed as *count* representing the score of danger. If a successful FCC request is received, the score is added by 1. Oppositely, it is subtracted by 1 if a failure is received.



3.3.2 Hardware Implementation

To implement TRW, we must track the establishment of FCC requests. It only considers the success or failure of connection attempts to new addresses. This approach inevitably requires a very large amount of state to keep track of which pairs of addresses have already tried to connect. When designing hardware, we often must store information in a fixed volume of memory. Since the information we would like to store may exceed this volume, one approach is to use approximate caches for which collisions will cause imperfections.

The scan detection and suppression algorithm approximates TRW in the following ways. The connections and addresses must be recorded using approximate caches. Figure 3.4 and 3.5 gives the overall data structures of the hardware implementation.

In Figure 3.4, connections are tracked using a fixed-sized table indexing by hashing the “inside” IP address, the “outside” IP address, and the inside port number for TCP. Each record consists of a 6-bit age counter and 1-bit field for each direction (connections from inside to outside and from outside to inside), recording whether a connection has been established in that direction.

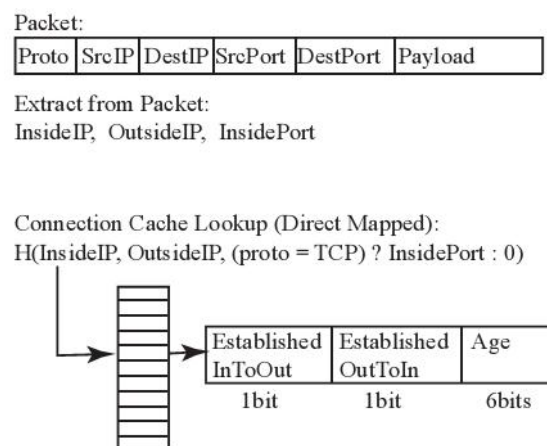


Figure 3.4 Connection cache

In Figure 3.5, external (outside) addresses are also tracked by an associative approximate cache. To find an entry, the external IP addresses are encrypted by a 32-bit block cipher. The resulting 32-bit number will be separated into an index and a tag. The index is used to find the line of entries. The “count” tracks the score of danger (add 1 when the connection is a failure and subtract 1 when it is successful).

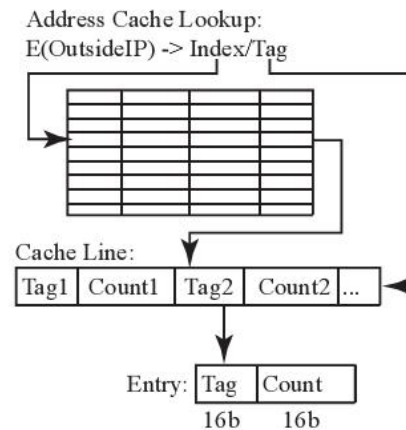


Figure 3.5 Address cache

3.3.3 Algorithm

Whenever the device receives a packet, it looks up the corresponding connection in the connection table and the corresponding external address in the address table. In Figure 3.6, the status of these two tables and the direction of the connection determine the action to take.

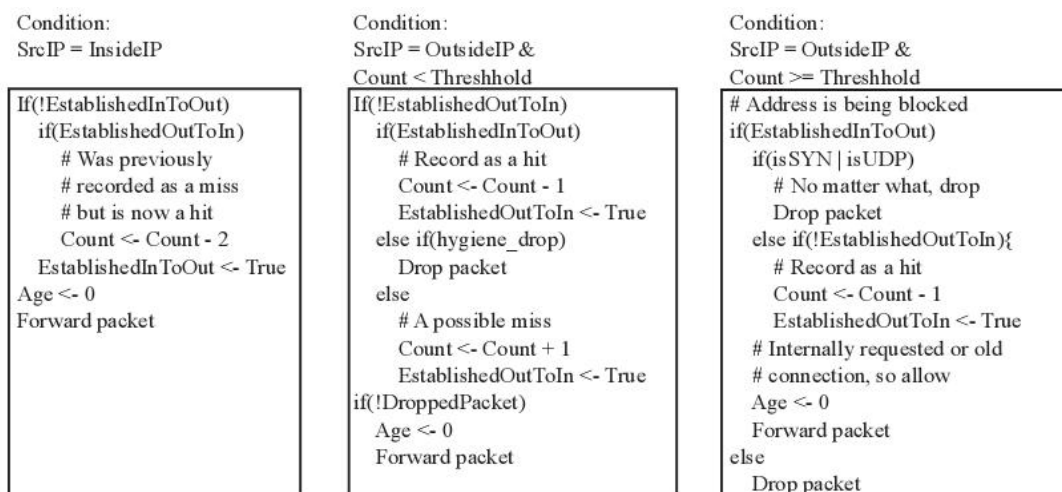


Figure 3.6 Algorithm for the simplified SHT

At first, consider the middle column of Figure 3.6. For a connection from a non-blocked outside IP address ($count < threshold$), reduce “age” to 0 and forward the packet if a corresponding connection has already been established in the packet’s direction. Otherwise, if the packet from the outside has been seen from the inside, forward the packet and decrement the address’s count by 1, as the outside address with a successful connection is credited. Otherwise, forward the packet but increment the address’s count by 1, as the address has one more outstanding, so-far unacknowledged connection request.

Likewise, for packets from inside addresses (the left column in Figure 3.6), if there is a connection establishment from the other direction, the count is reduced by 2 for compensation. This is because that the connection has been regarded as a failure previously and the count is added by 1.

Finally, consider the right column in Figure 3.6. If $count \geq threshold$, the device blocks it. When receiving subsequent packet from that address, the action depends on the packet’s type and whether it matches an existing and successfully established connection. If the packet does not match an existing connection, we drop it. If it does, then we will still drop it if it’s a UDP packet or a TCP initial SYN. Otherwise, we allow it through.

3.4 Reverse Sequential Hypothesis Testing

In the SHT, a host would no longer be observed when it was determined to be benign. In contrast, a scheme that concerned with detecting infection events is

proposed in the paper [3]. It is possible that a remote host is infected when its likelihood ratio is close to but larger than η_0 , as shown in Figure 3.7. In this case, it takes more observations for the SHT algorithm to declare it to be malicious than doing so for a host which is infected when its likelihood ratio is equal to 1.

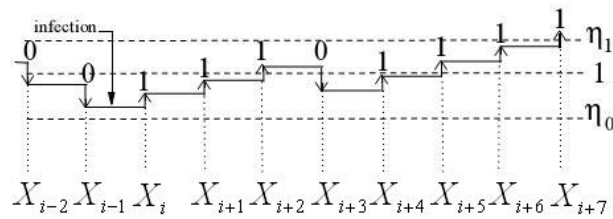


Figure 3.7 A log scale graph tracing the value of $\Lambda(\mathbf{X}_n)$ for SHT

3.4.1 Model

The solution to this problem is to run a new sequential hypothesis testing and evaluate the likelihood ratio in reverse chronological order when each connection is observed, as illustrated in Figure 3.8. To detect a host infected before Y_i but after Y_{i-1} , the reverse sequential hypothesis testing (RSHT) computes the likelihood ratio for the reversed vector of outcomes $\bar{\mathbf{X}}_n \equiv (X_n, \dots, X_1)$ observed so far. Because the most recent observations are process first, the RSHT will terminate before reaching the observations that were collected before infection.

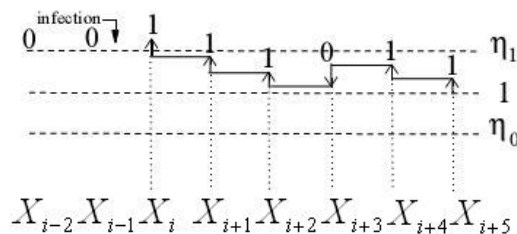


Figure 3.8 A log scale graph tracing the value of $\bar{\Lambda}(\mathbf{X}_n)$ for RSHT

A naïve implementation of repeated reverse sequential hypothesis testing requires storing an arbitrarily large sequence of FCC observation. In fact, there exists an iterative function with state variable $\bar{\Lambda}(\mathbf{X}_n)$ to optimize the computation.

$$\bar{\Lambda}(\mathbf{X}_n) = \max(1, \bar{\Lambda}(\mathbf{X}_{n-1})\phi(X_n)), \quad \bar{\Lambda}(\mathbf{X}_0) \equiv 1$$

It can be calculate in sequence when events are observed and maintain the likelihood ratio larger than one. Because $\bar{\Lambda}(\mathbf{X}_n)$ is updated in sequence, the observations can be discarded immediately after they are used to update $\bar{\Lambda}(\mathbf{X}_n)$.

3.4.2 Proof of Optimized Algorithm



The RSHT has the property that the likelihood value $\bar{\Lambda}(\mathbf{X}_n)$ of optimized computation exceed η_1 if and only if the RSHT starting backward from observation n concludes that the host was infected.

$$\bar{\Lambda}(\mathbf{X}_n) \geq \eta_1 \Leftrightarrow \Lambda(X_n, X_{n-1}, \dots, X_m) \geq \eta_1, \quad m \in [1, n] \quad \#$$

We first prove the following lemma starting if the RSHT reports an infection, the optimized algorithm will also report an infection.

Lemma 1:

For $\eta_1 > 1$ and for mutually independent random variables X_i ,
 $\forall m \in [1, n]: \Lambda(X_n, X_{n-1}, \dots, X_m) \geq \eta_1 \Rightarrow \bar{\Lambda}(\mathbf{X}_n) \geq \eta_1$

Proof:

We begin by replacing the Λ term with its equivalent expression in terms of ϕ .

$$\eta_1 \leq \Lambda(X_n, X_{n-1}, \dots, X_m) = \prod_{i=m}^n \phi(X_i)$$

We can place a lower bound on the value of $\bar{\Lambda}(\mathbf{X}_n)$ by exploiting the fact that, in any iteration, $\bar{\Lambda}$ can not return a value less than 1.

$$\bar{\Lambda}(\mathbf{X}_n) = \bar{\Lambda}(X_1, X_2, \dots, X_n) \geq 1 \times \bar{\Lambda}(X_m, X_{m+1}, \dots, X_n) \geq \prod_{i=m}^n \phi(X_i) \geq \eta_1$$

where the last inequality follows the steps taken in Equations. Thus,

$$\Lambda(X_n, X_{n-1}, \dots, X_m) \geq \eta_1 \Rightarrow \bar{\Lambda}(\mathbf{X}_n) \geq \eta_1 \quad \#$$

We must also prove that the optimized algorithm will only report an infection when the RSHT would also report an infection in reverse sequence. Recall that the RSHT will only report an infection if Λ exceeds η_1 before falling below η_0 .

Lemma 2:

For thresholds $\eta_0 < 1 < \eta_1$ and for mutually independent random variables X_i ,

if $\bar{\Lambda}(\mathbf{X}_i) \geq \eta_1$ for some $i = n$, but $\bar{\Lambda}(\mathbf{X}_i) < \eta_1$ for all $i \in [1, n-1]$, then there

(a) exists $m \in [1, n]$, such that $\Lambda(X_n, X_{n-1}, \dots, X_m) \geq \eta_1$

(b) exists no k in $[m, n]$, such that $\Lambda(X_n, X_{n-1}, \dots, X_k) \leq \eta_0$

Proof (a):

Find the largest m , such that $\bar{\Lambda}(\mathbf{X}_{m-2})\phi(X_{m-1}) < 1$.

It follows that $\bar{\Lambda}(\mathbf{X}_{m-1}) = 1$ and thus $\bar{\Lambda}(\mathbf{X}_m) = \bar{\Lambda}(\mathbf{X}_{m-1})\phi(X_m) = \phi(X_m)$.

Because we chose m such that $\bar{\Lambda}(\mathbf{X}_{j-2})\phi(X_{j-1}) \geq 1$ for all $j > m$, then

$$\bar{\Lambda}(\mathbf{X}_n) = \prod_{i=m}^n \phi(X_i) = \Lambda(X_n, X_{n-1}, \dots, X_m)$$

Thus, $\bar{\Lambda}(\mathbf{X}_n) \geq \eta_1 \Rightarrow \Lambda(X_n, X_{n-1}, \dots, X_m) \geq \eta_1$ #

Proof (b):

To prove that there exists no k in $[m, n]$ such that $\Lambda(X_n, X_{n-1}, \dots, X_k) \leq \eta_0$, suppose

that such a k exists. It follows that $\prod_{i=k}^n \phi(X_i) \leq \eta_0 < 1$.

Recall that we chose m to ensure that $\eta_1 \leq \prod_{i=m}^n \phi(X_i)$.

Separate the right hand side as follows:

$$\eta_1 \leq \prod_{i=m}^{k-1} \phi(X_i) \cdot \prod_{i=k}^n \phi(X_i) \leq \prod_{i=m}^{k-1} \phi(X_i) \leq \bar{\Lambda}(\mathbf{X}_{k-1})$$

This contradicts the assumption that $\bar{\Lambda}(\mathbf{X}_i) < \eta_1$ for all $i \in [1, n-1]$.

So there exists no k . #

3.4.3 Log of Likelihood Ratio

Similarly, the log of the likelihood ratio for RSHT can also be used to simplify

computation. The iterative function is equivalent to:

$$\bar{S}_n \equiv \ln(\bar{\Lambda}(\mathbf{X}_n)) = \max(0, \bar{S}_{n-1} + Y_n) = \begin{cases} \max(0, \bar{S}_{n-1} + F) = \bar{S}_{n-1} + F & \text{if } X_n = 1 \\ \max(0, \bar{S}_{n-1} + S) & \text{if } X_n = 0 \end{cases}$$

$$Y_n \equiv \ln(\phi(X_n)) = \begin{cases} \ln(1 - \theta_1) - \ln(1 - \theta_0) \equiv F > 0 & \text{if } X_n = 1 \\ \ln(\theta_1) - \ln(\theta_0) \equiv S < 0 & \text{if } X_n = 0 \end{cases}$$

To update the log of the likelihood ratio \bar{S}_n for each observation, addition and subtraction operations are adequate.



Chapter 4

Adaptive Sequential Hypothesis Testing

As mentioned in Chapter 3, the TRW algorithm assumes that θ_0 and θ_1 are known, which may not be true in a real network. According to the numerical results to be presented in Chapter 5, the false positive and false negative probabilities of the TRW algorithm could be much larger than the desired values if the adopted θ_0 and θ_1 are different from their true values. To overcome this problem, we propose in this chapter the adaptive algorithms to estimate the values of θ_0 and θ_1 based on observations of the outcomes of FCC attempts.

4.1 Scheme 1

Our proposed adaptive sequential hypothesis testing provides estimates of θ_0 and θ_1 adaptively based on observations of the outcomes of FCC attempts. The fixed values of θ_0 and θ_1 in the TRW algorithm are replaced with the variable estimates of $\hat{\theta}_0$ and $\hat{\theta}_1$ adaptively.

When a benign host r_i is detected, the value of $\hat{\theta}_0$ is updated using p_i , which is the success rate of FCC attempts sent by the detected benign host r_i . Let $N_i = S_i + F_i$ and $p_i = S_i/F_i$, where S_i and F_i represent, respectively, the numbers of successful and failed FCC attempts sent by r_i when it is detected as benign. Likewise, when a malicious host r_j is detected, the estimate $\hat{\theta}_1$ is updated by p_j , which is the success rate of FCC attempts sent by the detected malicious host r_j . The formulas can be shown as follows.

$$\hat{\theta}'_0 = \frac{m}{m+1} \hat{\theta}_0 + \frac{1}{m+1} p_i$$

$$\hat{\theta}'_1 = \frac{n}{n+1} \hat{\theta}_1 + \frac{1}{n+1} p_j$$

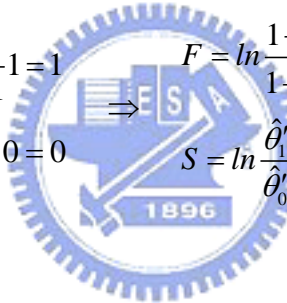
where m and n represent the numbers of benign and malicious hosts detected and adapted up to now, respectively. The next estimate $\hat{\theta}'_0$ is calculated according to the current estimates $\hat{\theta}_0$ and the success rate p_i of FCC attempts when a remote host r_i is newly determined as benign, and then the value of m is increased by 1. Likewise, the next estimate $\hat{\theta}'_1$ can be calculated according to $\hat{\theta}_1$ and p_j once a new malicious host r_j is detected, and then n value is increased by 1.

In the beginning, let $m=1$ and $n=1$. If $m=0$, the next estimate $\hat{\theta}'_0$ will equal to p_i when the first benign host is discovered. If $n=0$, the estimate $\hat{\theta}'_1$ will also become p_j when the first malicious host is discovered. Moreover, when the first few benign hosts are found, almost of FCC attempts sent by them are successful, i.e. the success rates p_i are nearly equal to 1. As long as the success

rate from the first benign host equals to 1, it will make $\hat{\theta}'_0 = 1$. Similarly, the first few detected malicious hosts have almost zero successful FCC attempts, such that the success rates p_j are nearly equal to 0. The estimate $\hat{\theta}'_1 = 0$ will happen once the success rates from the first malicious host equals to 0. The situations will cause that the step sizes of moving upward and downward become infinite.

As long as the success rate equals to 1 or it equals to 0, it may let $\hat{\theta}'_0 = 1$ or $\hat{\theta}'_1 = 0$. The situation will cause that the step sizes of moving upward and downward become infinite.

$$\hat{\theta}'_0 = \frac{0}{0+1}\hat{\theta}_0 + \frac{1}{0+1}1 = 1 \quad \Rightarrow \quad F = \ln \frac{1-\hat{\theta}'_1}{1-\hat{\theta}'_0} = \infty \quad (\text{upward})$$

$$\hat{\theta}'_1 = \frac{0}{0+1}\hat{\theta}_1 + \frac{1}{0+1}0 = 0 \quad \Rightarrow \quad S = \ln \frac{\hat{\theta}'_1}{\hat{\theta}'_0} = -\infty \quad (\text{downward})$$


Therefore, the adaptive formulas described above can be used to dynamically adjust the estimates of success rates conditioning on the benign and malicious hypotheses.

Because the earlier detected benign hosts will almost send successful FCC attempts, and the FCC attempts from the earlier detected malicious hosts will almost fail, the adaptive estimates may not be close to the real values if the adaptive procedure is performed in the beginning. So, we choose the duration in which the adaptive procedure is started. Let's define two parameters T_G and T_B , which denote the thresholds of benign (good) and malicious (bad) hosts. They are used to start the adaptive procedure when the number of the detected benign or malicious host

is more than T_G or T_B , respectively. At first, only the original TRW algorithm is implemented to examine FCC attempts sent by the remote hosts. As time goes by, it will detect N_G benign and N_B malicious hosts. When $N_G \geq T_G$, the adaptive procedure will be operated to update the new values of $\hat{\theta}'_0$ and m . Likewise, it will be operated to update the values of $\hat{\theta}'_1$ and n when $N_B \geq T_B$. The procedure will be stop if $|\hat{\theta}'_0 - \hat{\theta}_0| < \varepsilon$ or $|\hat{\theta}'_1 - \hat{\theta}_1| < \varepsilon$. Figure 4.1 shows the adaptive procedure.

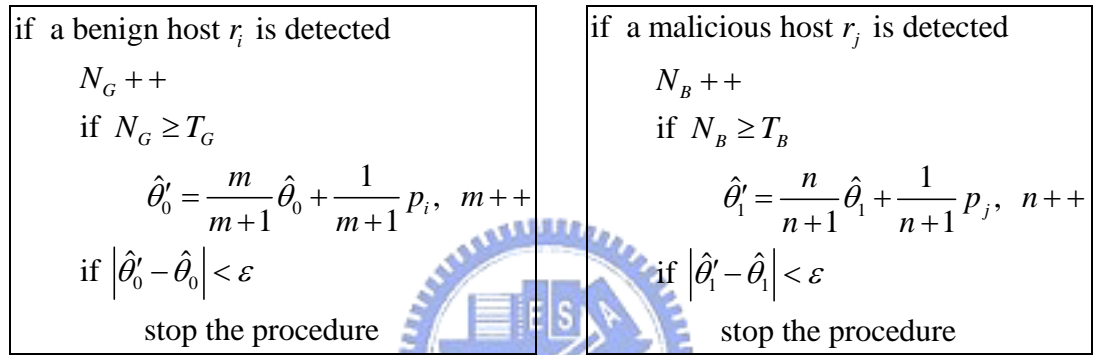


Figure 4.1 Adaptive procedure I

Suppose we can know the number of remote hosts that send connection attempts to local hosts and the ratio of benign to malicious hosts in advance. Therefore we can properly set the thresholds of T_G and T_B to start the adaptive procedure. For example, if there are 1000 remote hosts and the good-to-bad ratio is about 4:1, we can guess that there are approximately 800 benign hosts and 200 malicious hosts. The adaptive procedure will be started after a percentage of benign or malicious hosts are detected.

If $T_G = 0$ and $T_B = 0$, $\hat{\theta}'_0$ and $\hat{\theta}'_1$ will be update adaptively since the first benign host or malicious one is detected. If $T_G = 200$ and $T_B = 50$, it presents that

the procedure adjusts the two estimates after 25% of benign and malicious hosts are detected, respectively. We can also set the parameters to start the adaptive procedure only from 40% to 60% of hosts which is detected as benign or malicious. It just needs to set that the procedure will be activated when $320 \leq N_G \leq 480$ and $80 \leq N_B \leq 120$. We will show these simulation results in Chapter 5.

4.2 Scheme 2

If we only know how many hosts will send connection attempts rather than the good-to-bad ratio, we can calculate the ratio according to the number of hosts detected up to now. Suppose that there are totally N distinct remote hosts, and the TRW algorithm totally discovers N_G benign hosts and N_B malicious hosts so far. We can know that the good-to-bad ratio is $\frac{N_G}{N_G + N_B} : \frac{N_B}{N_G + N_B}$, and then we can approximate the expected values of benign and malicious hosts are $\frac{N_G}{N_G + N_B} \times N$ and $\frac{N_B}{N_G + N_B} \times N$, respectively.

Therefore, we can start the adaptive procedure from 40% to 60% of the expected numbers being detected. That is, the procedure will work when

$$\frac{N_G}{N_G + N_B} \times N \times 0.4 \leq N_G \leq \frac{N_G}{N_G + N_B} \times N \times 0.6$$

$$\frac{N_B}{N_G + N_B} \times N \times 0.4 \leq N_B \leq \frac{N_B}{N_G + N_B} \times N \times 0.6$$

Figure 4.2 shows the adaptive procedure. The simulate result will also be

presented in Chapter 5.

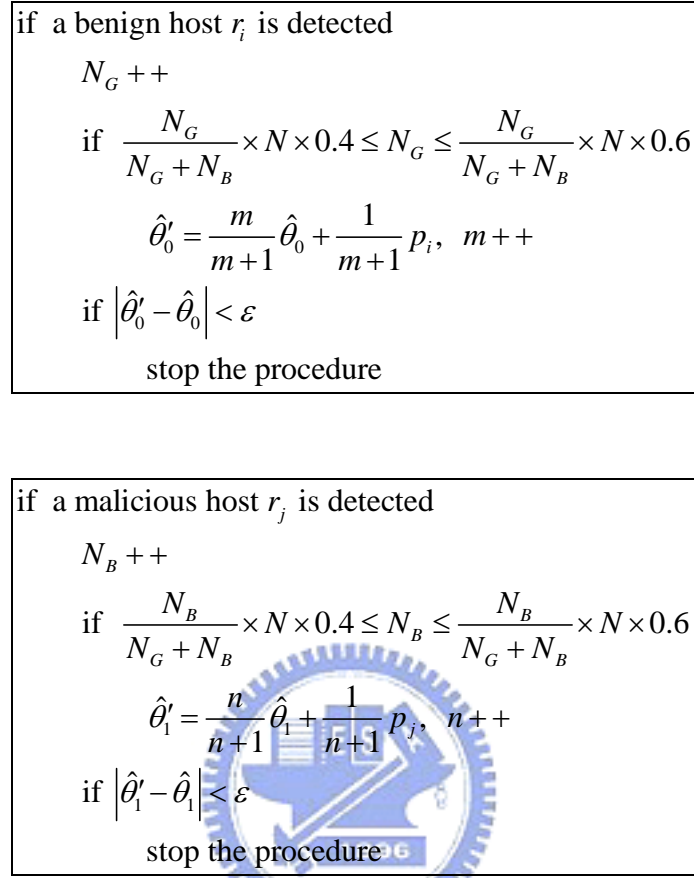


Figure 4.2 Adaptive procedure II

4.3 Implementation

Based on the hardware implementation introduced in Section 3.3.2, we propose a modified version of implementation. In order to perform SHT, the establishment of FCC requests must be tracked. Generally speaking, if a remote host r sends a connection request to a local host l , and then the host l replies an acknowledgement to the host r , the connection request is regarded as a success. Otherwise, it is a failure. For example, in Figure 4.3, the 1st and 4th connection requests are successful, but the

3rd, 6th, and 7th connection requests may be failed.

	source IP	destination IP	
No=1	138.230.222.48	140.113.173.14	← success
No=2	140.113.173.14	138.230.222.48	
No=3	146.80.28.246	140.113.135.13	← failure
No=4	123.26.187.115	140.113.134.229	← success
No=5	140.113.134.229	123.26.187.115	
No=6	128.16.190.43	140.113.159.237	← failure
No=7	162.241.77.123	140.113.96.240	← failure

Figure 4.3 List of connection

Each connection and the likelihood ratio of each remote host must be recorded. The IP addresses which send or receive connections can be classified as a remote IP of a local IP. A connection is tracked in a connection table indexing by hashing the remote IP address and the local IP address. Each record consists of a 1-bit field marking the connection from local to remote and the other 1-bit field marking the connection from remote to local. The former field set to 1 represents that the local host l has contacted with the remote host r , and the latter field set to 1 represents that the remote host r has contacted with the local host l . The 64-bit IP address is hashed to a 16-bit index, and the memory size of the connection cache is 128K bits. It is shown in Figure 4.4.

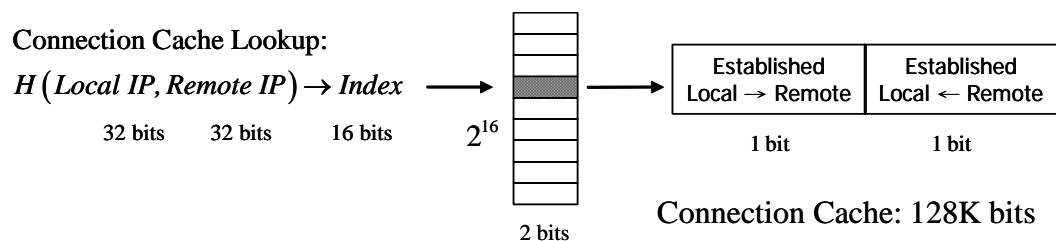


Figure 4.4 Connection Table

In Figure 4.5, the likelihood ratio of a remote IP address is also recorded in an address table indexing by hashing the remote IP address. Each entry records 16-bit likelihood ratio of the remote IP address. The 32-bit address is also hashed to a 16-bit index, and the memory usage of the address cache is 2M bits.

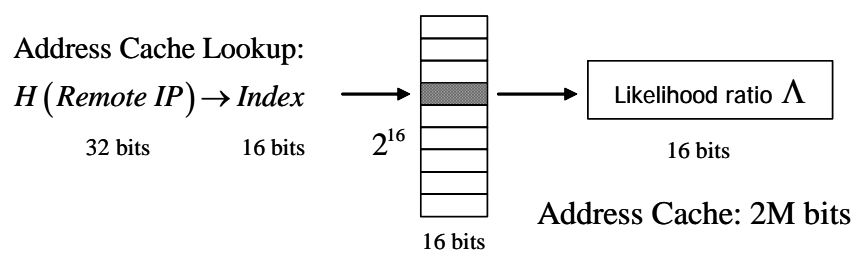


Figure 4.5 Address Table

When a connection is monitored by the scan detection machines, the detection mechanism looks up the connection in the connection table and the corresponding address of remote host in the address table. Then the modified algorithm based on SHT is performed to detect the remote host is benign or malicious. The algorithm is shown in Figure 4.6.

If a remote address r whose likelihood ratio is lower than the upper thresholds and higher than the lower bound sends a FCC request to a local address l , the connection request will be considered as a failure temporarily and the likelihood ratio of remote host r is updated as $\Lambda + F$. When a connection is sent by a local host l to a remote host r , if the host r has communicated with the host l , the connection request sent by host r previously must be a success. Therefore, the likelihood ratio of the

remote host r will be compensated such that it is updated as $\Lambda - F + S$.

Connection ($local \leftarrow remote$)

if $\frac{1-\beta}{1-\alpha} < \Lambda < \frac{\beta}{\alpha}$
 if $\rightarrow = 0$ & $\leftarrow = 0$
 $\Lambda = \Lambda + F$
 \leftarrow set to 1

Connection ($local \rightarrow remote$)

if $\rightarrow = 0$ & $\leftarrow = 1$
 $\Lambda = \Lambda - F + S$
 \rightarrow set to 1

if $\Lambda \geq \frac{\beta}{\alpha}$
malicious conclusion
 (*adaptive procedure*)

if $\Lambda \leq \frac{1-\beta}{1-\alpha}$
benign conclusion
 (*adaptive procedure*)

Figure 4.6 The modified algorithm for SHT

Chapter 5

Simulation Results

In this chapter, we will first present simulation results for the performances of the three detection algorithms introduced in Chapter 3, such as SHT, simplified SHT, and RSHT. The desired false positive rate and false negative rate are both assigned to 0.01. As a consequence, we choose $\alpha = 0.01$ and $\beta = 0.99$ in all simulations. Simulations are performed for 800 benign hosts and 200 malicious hosts.

We will compare the differences between known and unknown of the success rates θ_0 and θ_1 of connection attempts sent by benign hosts or malicious hosts. In a real network, θ_0 and θ_1 must be unknown but predictable adaptively. Then, we will also present simulation results for the performance of our proposed adaptive sequential hypothesis testing and compare with the previous algorithms.

5.1 SHT with known θ_0 and θ_1

In this section, we suppose that both the real values of θ_0 and θ_1 can be

known in advance. In each table shown below, we will orderly demonstrate 4 kinds of data, such as false positive rates (FP), false negative rates (FN), the average numbers of FCC attempts sent by a benign host before being detected (N_G), and the average numbers of FCC attempts sent by a malicious host before being detected (N_B). The horizontal axle represents various values of θ_0 , and the vertical axle represents various values of θ_1 .

At first, we show Table 5.1 which represents the step sizes of moving upward and downward for SHT as the values of θ_0 and θ_1 are changed. It tells that the higher success rate of the benign hypothesis θ_0 leads to the larger step size of moving upward, and the lower success rate of the malicious hypothesis θ_1 leads to the larger step size of moving downward.

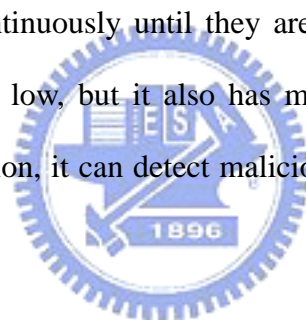
Table 5.2 shows the results of SHT algorithm for the combinations of θ_0 and θ_1 , assuming that they are known in advance. As one can see, the false positive and false negative probabilities are close to the desired values 0.01. The values of N_G and N_B , average numbers of FCC attempts sent by a benign and malicious host before detected, are small, especially when θ_0 is larger and θ_1 is smaller.

Table 5.3 shows the results of the simplified SHT. The step sizes of moving upward and downward for SHT are changed according the value of θ_0 and θ_1 , but the step sizes for simplified SHT are fixed value. So, we can regard the simplified SHT as a special case of the original SHT. In the tables, we will find the phenomena that the false positive rates increase when θ_0 is small, and the false negative rates increase when θ_1 is large. For the original SHT, the step size of moving upward must be increasing when θ_0 is increasing, and the step size of moving downward must be increasing when θ_1 is decreasing. Therefore, when θ_0 is smaller and θ_1

is larger, the step sizes for the simplified SHT are both larger than those for the original SHT. It will exceed the thresholds easily and cause more and more false positives and false negatives.

Because of the step sizes of moving upward and downward, the average numbers of FCC attempts before detected are also different from those of SHT. When there are larger θ_0 and smaller θ_1 , the step sizes of simplified SHT are smaller than those of SHT, so the value of N_G and N_B are larger. Similarly, the values of observation are smaller when θ_0 is smaller and θ_1 is larger.

Table 5.4 is the result of RSHT. Because it only detect the malicious hosts and monitor the benign hosts continuously until they are infected, we can find that the false negative rates are quite low, but it also has much higher false positive rates. Because of the reverse detection, it can detect malicious hosts slightly faster than the SHT algorithm.



Failure	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.747	0.865	0.999	1.153	1.335	1.558	1.846	2.251	2.944
0.10	0.693	0.811	0.944	1.099	1.281	1.504	1.792	2.197	2.890
0.15	0.636	0.754	0.887	1.041	1.224	1.447	1.735	2.140	2.833
0.20	0.575	0.693	0.827	0.981	1.163	1.386	1.674	2.079	2.773
0.25	0.511	0.629	0.762	0.916	1.099	1.322	1.609	2.015	2.708
0.30	0.442	0.560	0.693	0.847	1.030	1.253	1.540	1.946	2.639
0.35	0.368	0.486	0.619	0.773	0.956	1.179	1.466	1.872	2.565
0.40	0.288	0.405	0.539	0.693	0.875	1.099	1.386	1.792	2.485
0.45	0.201	0.318	0.452	0.606	0.788	1.012	1.299	1.705	2.398
Success	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	-2.398	-2.485	-2.565	-2.639	-2.708	-2.773	-2.833	-2.890	-2.944
0.10	-1.705	-1.792	-1.872	-1.946	-2.015	-2.079	-2.140	-2.197	-2.251
0.15	-1.299	-1.386	-1.466	-1.540	-1.609	-1.674	-1.735	-1.792	-1.846
0.20	-1.012	-1.099	-1.179	-1.253	-1.322	-1.386	-1.447	-1.504	-1.558
0.25	-0.788	-0.875	-0.956	-1.030	-1.099	-1.163	-1.224	-1.281	-1.335
0.30	-0.606	-0.693	-0.773	-0.847	-0.916	-0.981	-1.041	-1.099	-1.153
0.35	-0.452	-0.539	-0.619	-0.693	-0.762	-0.827	-0.887	-0.944	-0.999
0.40	-0.318	-0.405	-0.486	-0.560	-0.629	-0.693	-0.754	-0.811	-0.865
0.45	-0.201	-0.288	-0.368	-0.442	-0.511	-0.575	-0.636	-0.693	-0.747

Table 5.1 The step sizes of failure and success for SHT



FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.64	0.67	0.72	0.96	0.55	0.95	0.47	0.29	0.46
0.10	0.83	0.89	0.96	0.68	0.68	0.46	0.67	0.31	0.43
0.15	0.71	0.86	0.75	0.88	0.71	0.51	0.70	0.61	0.50
0.20	0.91	0.82	0.87	0.84	0.80	0.62	0.78	0.59	0.52
0.25	0.86	0.80	0.81	0.73	0.54	0.70	0.76	0.62	0.55
0.30	0.76	0.86	0.91	0.73	0.76	0.67	0.78	0.64	0.63
0.35	0.55	0.74	0.81	0.89	0.86	0.75	0.73	0.81	0.68
0.40	0.23	0.48	0.75	0.81	0.91	0.82	0.82	0.72	0.72
0.45	0.13	0.19	0.54	0.71	0.78	0.80	0.78	0.71	0.75
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	1.34	1.43	1.36	1.20	1.29	1.08	1.16	1.08	0.87
0.10	1.14	1.26	1.36	1.29	1.36	1.10	1.12	0.72	0.70
0.15	1.38	1.28	1.15	1.32	1.37	1.13	1.23	0.98	0.83
0.20	1.35	1.38	1.27	1.08	1.10	1.16	0.97	0.83	1.43
0.25	1.31	1.37	1.41	1.24	1.02	1.21	1.04	0.99	0.86
0.30	1.08	1.22	1.34	1.11	1.14	1.32	1.27	1.04	1.31
0.35	0.74	1.07	1.19	1.30	1.07	1.33	1.21	1.36	1.13
0.40	0.38	0.72	1.08	1.40	1.33	1.35	1.11	1.24	0.90
0.45	0.00	0.30	0.69	1.07	1.23	1.30	1.21	1.07	0.85
N G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	5.51	4.74	4.17	3.78	3.39	2.98	2.71	2.48	2.21
0.10	8.38	7.21	5.69	5.24	4.60	4.12	3.70	3.76	3.32
0.15	11.89	9.59	8.09	6.64	5.65	4.95	4.25	3.75	3.48
0.20	16.01	13.04	10.59	8.90	7.54	6.63	5.68	5.00	3.51
0.25	21.30	16.99	13.59	11.51	9.89	7.59	6.43	5.46	4.69
0.30	27.03	22.19	17.80	14.54	12.01	9.10	7.62	6.49	4.90
0.35	33.51	27.98	22.90	18.01	14.83	11.28	9.44	7.08	6.15
0.40	40.95	34.41	28.22	22.90	18.34	14.32	11.24	8.80	7.45
0.45	61.29	41.56	34.43	28.38	22.80	17.79	14.27	10.89	8.70
N B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	8.74	7.44	6.12	4.92	4.72	3.53	3.49	3.33	2.22
0.10	10.85	8.84	7.08	6.54	5.46	5.03	3.76	3.77	2.51
0.15	14.34	11.29	9.43	7.62	6.52	5.69	4.27	3.71	2.73
0.20	17.90	14.32	11.35	9.21	7.65	6.70	5.04	4.16	3.01
0.25	22.68	18.43	14.95	12.08	9.96	7.60	5.72	4.65	3.43
0.30	28.43	22.92	18.14	14.58	11.65	8.94	6.69	5.27	3.81
0.35	34.39	28.23	23.05	17.74	13.79	10.64	8.21	5.71	4.21
0.40	41.34	34.47	28.02	22.44	17.04	13.15	9.76	7.25	4.79
0.45	47.80	40.19	33.22	26.96	21.43	16.23	11.91	8.39	5.58

Table 5.2 SHT, θ_0 and θ_1 are known

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	24.70	11.52	4.56	1.69	0.68	0.34	0.22	0.19	0.16
0.10	24.76	11.50	4.42	1.72	0.69	0.32	0.21	0.17	0.11
0.15	25.00	11.37	4.52	1.77	0.62	0.30	0.19	0.11	0.08
0.20	24.64	11.27	4.38	1.59	0.60	0.29	0.16	0.08	0.07
0.25	24.38	11.33	4.45	1.65	0.54	0.21	0.10	0.06	0.03
0.30	24.66	11.46	4.43	1.57	0.55	0.17	0.06	0.02	0.02
0.35	24.51	11.24	4.57	1.53	0.49	0.11	0.05	0.02	0.01
0.40	24.68	11.21	4.48	1.49	0.47	0.13	0.03	0.02	0.01
0.45	24.52	11.24	4.30	1.49	0.42	0.12	0.03	0.01	0.01
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.01	0.02	0.09	0.09	0.18	0.21	0.38	0.57	0.60
0.10	0.04	0.07	0.10	0.17	0.28	0.38	0.40	0.64	0.77
0.15	0.07	0.12	0.18	0.20	0.35	0.43	0.65	0.78	0.93
0.20	0.18	0.35	0.38	0.34	0.56	0.83	0.79	0.93	1.07
0.25	0.52	0.67	0.83	0.85	1.02	1.13	1.20	1.33	1.39
0.30	1.75	1.76	1.96	1.94	2.21	2.27	2.42	2.46	2.60
0.35	4.46	4.71	4.83	4.95	5.23	5.16	5.51	5.30	5.55
0.40	11.49	12.15	12.16	12.30	11.98	12.12	12.20	12.41	12.03
0.45	24.29	24.74	25.06	25.15	25.17	25.83	25.48	24.49	25.45
N G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	19.12	17.14	14.62	12.02	9.91	8.32	7.16	6.29	5.59
0.10	19.08	17.28	14.59	12.01	9.91	8.32	7.16	6.28	5.59
0.15	19.12	17.16	14.59	12.03	9.89	8.35	7.16	6.28	5.59
0.20	19.06	17.25	14.60	12.05	9.92	8.36	7.17	6.28	5.59
0.25	19.11	17.19	14.63	12.01	9.89	8.34	7.18	6.28	5.59
0.30	19.11	17.11	14.67	12.01	9.93	8.32	7.18	6.27	5.60
0.35	19.00	17.21	14.57	12.01	9.97	8.35	7.17	6.28	5.59
0.40	19.09	17.19	14.62	12.03	9.96	8.35	7.16	6.28	5.59
0.45	19.07	17.15	14.63	12.01	9.94	8.33	7.16	6.27	5.58
N B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	5.64	5.66	5.65	5.70	5.73	5.72	5.70	5.69	5.67
0.10	6.31	6.34	6.38	6.42	6.42	6.36	6.38	6.36	6.35
0.15	7.24	7.20	7.24	7.26	7.34	7.25	7.25	7.22	7.18
0.20	8.48	8.40	8.42	8.47	8.42	8.45	8.48	8.39	8.36
0.25	9.98	10.08	10.09	10.02	9.96	10.01	9.98	9.95	9.93
0.30	12.07	12.17	12.07	12.04	12.13	12.09	12.09	12.07	11.93
0.35	14.54	14.60	14.74	14.64	14.71	14.52	14.73	14.55	14.59
0.40	17.15	17.16	17.06	17.30	17.17	17.20	17.33	17.18	17.16
0.45	19.04	19.16	18.95	18.87	19.06	19.10	19.18	19.14	19.04

Table 5.3 Simplified SHT, θ_0 and θ_1 are known

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	10.40	11.70	15.54	24.48	13.90	27.43	13.34	4.97	11.53
0.10	10.81	13.49	17.22	11.97	14.22	8.20	14.90	4.97	11.39
0.15	7.21	9.74	9.85	12.36	14.19	8.35	14.94	11.40	11.80
0.20	7.44	8.07	10.57	12.45	14.31	8.52	14.90	11.49	11.74
0.25	4.86	5.98	6.42	6.44	6.39	9.25	15.15	11.38	11.78
0.30	3.04	4.66	6.61	6.53	7.21	9.22	14.90	11.38	11.92
0.35	1.66	2.93	4.11	6.73	8.01	9.13	9.07	12.28	12.44
0.40	0.52	1.39	2.84	4.29	6.47	7.15	9.04	8.15	12.80
0.45	0.04	0.48	1.67	3.07	4.47	5.85	6.58	8.16	12.71
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.03	0.02	0.02	0.00	0.01	0.00	0.00	0.00	0.00
0.10	0.03	0.01	0.02	0.02	0.01	0.00	0.00	0.00	0.00
0.15	0.18	0.03	0.02	0.01	0.00	0.00	0.00	0.00	0.00
0.20	0.73	0.20	0.03	0.03	0.01	0.01	0.00	0.00	0.00
0.25	3.83	0.96	0.28	0.06	0.03	0.01	0.00	0.00	0.00
0.30	13.56	3.99	0.98	0.29	0.06	0.02	0.01	0.00	0.00
0.35	37.53	14.96	4.37	1.01	0.14	0.06	0.01	0.00	0.00
0.40	76.13	40.46	13.89	3.94	0.87	0.14	0.03	0.00	0.00
0.45	98.45	75.08	36.38	12.57	3.30	0.59	0.10	0.00	0.01
N G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
N B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	8.56	7.26	5.96	4.69	4.67	3.42	3.42	3.38	2.21
0.10	10.06	8.26	6.71	6.39	5.25	4.99	3.68	3.68	2.40
0.15	13.32	10.54	8.86	7.20	5.99	5.53	4.06	3.69	2.60
0.20	16.19	13.05	10.32	8.32	6.80	6.29	4.61	4.01	2.83
0.25	21.02	16.90	13.71	11.31	9.18	7.05	5.14	4.39	3.11
0.30	27.16	21.39	16.26	13.27	10.63	8.13	5.82	4.85	3.41
0.35	33.67	26.96	21.56	15.86	12.32	9.42	7.55	5.29	3.76
0.40	41.11	33.76	26.86	20.85	15.33	11.81	8.65	6.63	4.20
0.45	50.45	40.28	32.65	25.67	19.70	14.69	10.84	7.52	4.73

Table 5.4 RSHT, θ_0 and θ_1 are known

5.2 SHT with unknown θ_0 and θ_1

In fact, the values of θ_0 and θ_1 must be unknown in advance. If we use the different values from the real ones to detect the hosts, the detection results must be affected. Subsequently, we guess that $\hat{\theta}_0 = 0.8$ and $\hat{\theta}_1 = 0.2$ and simulate both SHT and RSHT with unknown θ_0 and θ_1 . The simplified SHT will be ignored because it can't be affected by various values of θ_0 and θ_1 .

Table 5.5 shows the results of the SHT algorithm with unknown θ_0 and θ_1 . Compared with Table 5.2, the false positive and negative rates are both much higher. There are similar results for the RSHT algorithm shown in Table 5.6. These results indicate that the erroneous estimated values of θ_0 and θ_1 will cause the erroneous detection. In the next section, we will implement the original SHT with our proposed adaptive estimation algorithm introduced in the previous chapter and then compare the simulation results with the SHT without adaptive estimation.

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	30.46	16.75	8.03	3.50	1.45	0.65	0.31	0.21	0.17
0.10	30.52	16.55	7.96	3.59	1.45	0.60	0.24	0.19	0.11
0.15	30.36	16.65	7.87	3.48	1.48	0.57	0.25	0.12	0.11
0.20	30.53	16.58	8.05	3.37	1.29	0.54	0.23	0.13	0.07
0.25	30.38	16.45	7.95	3.43	1.38	0.52	0.15	0.09	0.04
0.30	30.38	16.58	8.02	3.44	1.27	0.50	0.17	0.05	0.04
0.35	30.72	16.50	7.89	3.53	1.29	0.43	0.14	0.05	0.03
0.40	30.27	16.43	8.06	3.31	1.31	0.44	0.15	0.03	0.02
0.45	30.38	16.68	7.66	3.38	1.31	0.46	0.12	0.03	0.01
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.04	0.07	0.06	0.12	0.18	0.31	0.33	0.51	0.62
0.10	0.14	0.10	0.12	0.23	0.27	0.37	0.54	0.63	0.73
0.15	0.20	0.30	0.35	0.36	0.45	0.67	0.73	0.80	0.99
0.20	0.54	0.59	0.69	0.80	0.82	0.87	1.09	1.21	1.20
0.25	1.58	1.39	1.52	1.78	1.78	1.94	1.85	2.01	2.17
0.30	3.42	3.64	3.58	3.69	4.05	4.16	4.11	4.23	4.08
0.35	8.27	8.23	7.49	8.12	8.10	8.69	8.65	8.70	8.77
0.40	16.42	16.70	16.63	16.79	17.06	16.81	17.24	17.80	16.89
0.45	30.77	30.32	30.64	30.83	31.19	31.27	31.53	31.35	31.22
N G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	14.27	12.94	11.11	9.32	7.80	6.60	5.72	5.02	4.46
0.10	14.23	12.90	11.11	9.33	7.80	6.61	5.73	5.01	4.47
0.15	14.29	12.92	11.10	9.34	7.83	6.62	5.71	5.02	4.47
0.20	14.34	12.89	11.10	9.34	7.82	6.63	5.72	5.01	4.47
0.25	14.30	12.90	11.07	9.34	7.81	6.65	5.71	5.04	4.47
0.30	14.24	12.90	11.14	9.31	7.83	6.63	5.71	5.02	4.46
0.35	14.25	12.85	11.06	9.34	7.82	6.63	5.71	5.02	4.46
0.40	14.26	12.94	11.10	9.33	7.79	6.64	5.72	5.02	4.46
0.45	14.28	12.99	11.14	9.34	7.82	6.63	5.72	5.01	4.46
N B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	4.49	4.53	4.52	4.54	4.53	4.52	4.55	4.54	4.51
0.10	5.04	5.09	5.09	5.09	5.08	5.08	5.03	5.07	5.04
0.15	5.78	5.76	5.76	5.78	5.80	5.79	5.74	5.76	5.75
0.20	6.69	6.66	6.66	6.70	6.71	6.67	6.66	6.71	6.65
0.25	7.90	7.94	7.83	7.92	7.85	7.85	7.87	7.84	7.82
0.30	9.37	9.29	9.34	9.38	9.38	9.36	9.31	9.27	9.37
0.35	11.11	11.13	11.19	11.13	11.24	11.12	11.11	11.07	11.04
0.40	13.02	12.84	13.05	12.94	12.84	12.98	13.02	12.91	12.91
0.45	14.18	14.31	14.29	14.36	14.33	14.28	14.21	14.19	14.17

Table 5.5 SHT, θ_0 and θ_1 unknown, guess 0.8 and 0.2

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	85.75	71.42	53.41	34.89	18.85	8.52	2.85	0.85	0.35
0.10	85.51	71.46	52.96	34.56	18.89	8.26	2.95	0.83	0.32
0.15	85.50	71.71	53.12	34.35	19.05	8.40	3.03	0.81	0.35
0.20	85.57	71.22	53.02	34.50	18.73	8.42	2.95	0.81	0.31
0.25	85.32	71.38	53.19	34.58	18.92	8.50	2.83	0.82	0.31
0.30	85.56	71.60	53.28	34.41	18.77	8.44	2.85	0.81	0.25
0.35	85.44	71.45	53.47	34.55	18.73	8.34	3.02	0.74	0.25
0.40	85.59	71.42	53.30	34.16	19.04	8.17	2.92	0.75	0.25
0.45	85.41	71.57	53.15	34.61	18.47	8.39	2.89	0.76	0.22
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
0.10	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02
0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02
0.20	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.04
0.25	0.01	0.00	0.01	0.00	0.00	0.01	0.03	0.05	0.09
0.30	0.01	0.01	0.02	0.01	0.01	0.04	0.04	0.09	0.19
0.35	0.09	0.10	0.14	0.10	0.11	0.13	0.18	0.25	0.33
0.40	0.51	0.43	0.52	0.51	0.48	0.58	0.66	0.70	0.81
0.45	1.84	2.01	1.99	2.02	2.00	2.23	2.10	2.24	2.59
N G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
N B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	4.42	4.45	4.44	4.46	4.47	4.51	4.51	4.56	4.59
0.10	4.88	4.92	4.92	4.94	4.94	4.96	4.97	5.04	5.05
0.15	5.48	5.48	5.48	5.49	5.53	5.57	5.57	5.62	5.67
0.20	6.19	6.16	6.19	6.24	6.26	6.27	6.28	6.37	6.38
0.25	7.12	7.11	7.08	7.19	7.17	7.17	7.20	7.25	7.38
0.30	8.23	8.21	8.26	8.29	8.40	8.43	8.47	8.47	8.45
0.35	9.78	9.84	9.82	9.86	9.94	10.05	9.93	9.98	10.01
0.40	11.86	11.89	11.94	11.93	11.95	12.03	12.08	12.13	11.99
0.45	14.25	14.40	14.40	14.62	14.67	14.55	14.61	14.60	14.52

Table 5.6 RSHT, θ_0 and θ_1 unknown, guess 0.8 and 0.2

5.3 Adaptive Sequential Hypothesis Testing

In this section, we simulate our proposed adaptive sequential hypothesis testing and compare the performance with SHT without adaptive estimation. At first, we update the values of $\hat{\theta}_0$ and $\hat{\theta}_1$ using the formula introduced in Section 4.1 after the first benign or malicious host is detected. In the other word, when $N_G \geq 0$ or $N_B \geq 0$, the values of θ_0 and θ_1 are updated. Table 5.8 shows the simulation results. Compare with the Table 5.5 setting $\theta_0 = 0.8$ and $\theta_1 = 0.2$, most of the false positive and negative rates become a little lower.

Table 5.9 shows the results of updating estimates after 25% of benign or malicious hosts are detected. It represents that the success rates of the first few detected hosts will be different from the expected values. Without updating the first 25% of hosts, the false positives and negatives are less than the former.

Therefore, we further simulate the scheme that update the estimates from 40% to 60% of remote hosts detected as benign or malicious. When $320 \leq N_G \leq 480$ or $80 \leq N_B \leq 120$, the adaptive procedure is activated. In Table 5.10, we find that most of the false positive and negative rates are reduced to 2~3%

In fact, we don't know the exact numbers of benign and malicious hosts. As described in Section 4.2, we can only predict them according to the good-to-bad ratio. When the numbers of detected benign and malicious hosts are located between 40% and 60% of the expected numbers, the procedure starts. Table 5.11 shows that the scheme can reduce the probabilities and false positive and negative. Table 5.12 shows that the values of θ_0 and θ_1 are close to the real values finally.

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	26.52	20.99	16.33	12.75	8.54	6.29	4.39	2.92	2.51
0.10	28.60	21.92	17.12	13.41	9.68	6.50	4.69	3.40	2.44
0.15	31.03	23.84	17.98	14.50	10.23	7.24	4.71	3.32	2.35
0.20	32.93	25.94	20.10	15.33	10.62	7.64	5.03	3.20	2.49
0.25	35.18	28.42	21.18	15.79	11.83	8.26	5.38	3.78	2.43
0.30	38.35	28.49	22.61	17.28	12.55	8.82	5.80	3.50	2.33
0.35	39.15	30.87	23.26	18.56	14.13	9.67	6.15	3.83	2.43
0.40	39.20	32.63	25.46	18.67	14.29	9.86	6.56	3.71	2.34
0.45	40.55	33.22	26.99	20.97	14.86	10.54	6.58	4.27	2.41
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.88	0.73	0.78	0.90	1.06	1.06	1.48	1.50	1.74
0.10	1.77	1.85	2.15	1.99	2.06	1.98	2.16	2.63	2.74
0.15	3.50	3.57	3.49	3.38	3.53	3.40	3.69	3.90	4.22
0.20	5.84	5.45	5.79	5.16	5.27	5.14	5.47	5.70	5.73
0.25	8.92	8.33	8.53	7.95	7.52	7.41	7.56	7.72	8.28
0.30	13.14	12.84	11.92	10.70	10.14	10.68	9.80	10.35	10.59
0.35	19.36	17.18	16.50	14.32	13.32	12.77	13.20	13.23	13.54
0.40	26.67	24.56	20.69	20.92	18.11	17.41	16.67	16.92	16.01
0.45	35.23	32.64	29.06	26.19	25.02	22.24	21.76	20.90	20.92
N _G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	4.99	4.51	4.03	3.61	3.22	2.91	2.60	2.32	2.07
0.10	5.37	4.78	4.27	3.85	3.44	3.09	2.80	2.47	2.20
0.15	5.73	5.22	4.65	4.17	3.72	3.31	2.94	2.58	2.33
0.20	5.90	5.51	4.99	4.50	3.95	3.50	3.15	2.76	2.43
0.25	6.18	5.80	5.26	4.66	4.16	3.71	3.31	2.87	2.49
0.30	6.26	5.96	5.52	5.02	4.50	3.97	3.49	3.02	2.59
0.35	6.41	6.25	5.70	5.32	4.81	4.22	3.71	3.16	2.69
0.40	6.68	6.26	5.98	5.36	5.02	4.44	3.84	3.31	2.82
0.45	6.72	6.44	6.13	5.67	5.15	4.59	4.01	3.42	2.92
N _B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	2.86	2.76	2.65	2.55	2.51	2.38	2.29	2.18	1.84
0.10	3.22	3.10	2.95	2.83	2.73	2.62	2.50	2.30	1.99
0.15	3.58	3.49	3.33	3.13	2.99	2.82	2.68	2.51	2.18
0.20	3.99	3.83	3.66	3.43	3.31	3.14	2.95	2.75	2.29
0.25	4.49	4.24	4.01	3.85	3.56	3.37	3.16	2.88	2.46
0.30	4.92	4.84	4.49	4.25	3.99	3.73	3.43	3.14	2.67
0.35	5.58	5.30	5.10	4.70	4.34	4.04	3.82	3.37	2.85
0.40	6.25	5.80	5.53	5.22	4.92	4.53	4.12	3.70	3.15
0.45	6.72	6.39	6.07	5.62	5.41	4.88	4.55	4.00	3.35

Table 5.8 Adaptive SHT, $N_G \geq 0$ & $N_B \geq 0$ (0%)

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	7.43	6.33	5.45	4.52	3.40	3.06	2.44	1.43	1.23
0.10	8.91	6.77	6.26	5.21	4.78	3.89	2.95	1.98	1.35
0.15	10.12	7.59	6.53	5.06	4.19	3.81	2.91	2.21	1.58
0.20	11.71	8.72	7.12	5.24	3.98	3.65	2.62	1.80	1.62
0.25	13.00	10.44	7.62	5.71	4.87	3.86	2.68	1.93	1.19
0.30	15.57	10.86	8.55	6.60	5.65	4.47	3.05	2.19	1.21
0.35	16.74	12.60	9.56	7.15	6.37	5.03	3.39	2.78	1.36
0.40	18.39	13.94	10.59	8.25	6.81	4.91	3.69	2.91	1.61
0.45	20.33	14.80	11.63	9.24	7.18	5.55	3.96	3.18	1.68
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	1.31	1.33	1.61	1.25	1.24	1.62	1.59	1.43	1.15
0.10	2.15	2.38	2.77	2.34	2.02	1.87	2.29	2.04	1.68
0.15	3.51	4.66	4.48	3.49	2.99	2.93	3.52	2.89	2.81
0.20	6.07	6.28	6.37	4.88	3.90	3.94	4.54	3.58	4.07
0.25	9.22	8.58	8.76	6.98	4.77	4.99	5.56	4.15	4.97
0.30	13.83	12.67	12.40	9.04	6.48	6.83	6.55	5.53	5.70
0.35	21.40	17.45	16.51	11.53	8.22	8.17	8.74	6.23	6.89
0.40	30.69	25.46	21.76	16.31	11.57	10.67	10.05	7.45	7.62
0.45	42.93	37.48	30.27	22.24	16.06	13.52	12.64	8.98	10.28
N _G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	5.12	4.25	3.57	3.44	3.18	2.76	2.44	2.26	2.13
0.10	6.18	5.07	4.24	3.94	3.60	3.13	2.66	2.45	2.26
0.15	7.24	5.83	4.89	4.59	4.14	3.56	2.91	2.71	2.37
0.20	8.12	6.82	5.66	5.18	4.77	4.07	3.44	3.26	2.61
0.25	9.38	7.84	6.27	5.72	5.43	4.52	3.87	3.65	2.97
0.30	10.50	8.83	7.13	6.56	6.11	5.08	4.32	3.89	3.25
0.35	11.23	9.88	8.01	7.51	6.96	5.71	4.72	4.33	3.46
0.40	12.30	10.81	9.05	8.26	7.68	6.43	5.26	4.83	3.74
0.45	13.04	11.42	10.28	9.35	8.55	7.13	5.90	5.40	4.03
N _B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	4.68	4.17	3.65	3.40	3.26	2.85	2.41	2.25	2.04
0.10	5.50	4.94	4.23	3.86	3.44	3.04	2.64	2.39	2.13
0.15	6.43	5.75	4.94	4.46	4.05	3.46	2.93	2.61	2.24
0.20	7.43	6.76	5.84	5.22	4.79	4.09	3.49	3.09	2.42
0.25	8.90	7.86	6.75	6.06	5.26	4.54	3.96	3.45	2.79
0.30	9.95	9.21	7.90	6.95	5.96	5.13	4.35	3.72	3.04
0.35	11.16	10.40	9.07	8.38	6.84	5.76	4.95	3.98	3.31
0.40	11.84	11.58	10.26	9.47	8.25	6.93	5.65	4.52	3.55
0.45	12.08	11.90	11.45	10.87	9.53	7.87	6.52	5.22	3.91

Table 5.9 Adaptive SHT, $N_G \geq 200$ & $N_B \geq 50$ (25%)

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	3.48	3.52	3.77	3.09	2.31	2.17	2.02	1.28	0.84
0.10	3.46	3.13	2.93	2.60	2.68	2.00	2.13	1.57	0.93
0.15	3.48	3.16	3.12	2.60	2.31	1.89	2.08	1.34	0.92
0.20	3.61	3.26	3.00	2.57	2.38	2.03	1.70	1.33	0.87
0.25	3.76	3.59	2.92	2.87	2.42	2.26	1.88	1.57	0.87
0.30	4.22	3.83	3.11	3.03	2.63	2.28	1.75	1.55	0.86
0.35	4.67	4.03	3.57	3.03	2.73	2.49	1.87	1.63	0.89
0.40	4.77	4.03	3.53	3.26	2.94	2.55	2.07	1.71	1.00
0.45	5.31	4.07	3.56	3.37	3.09	2.86	2.16	1.76	1.05
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	1.62	1.24	1.54	1.22	1.50	1.30	1.34	1.34	1.01
0.10	2.17	1.99	2.22	1.87	2.19	1.76	1.81	2.07	1.54
0.15	2.74	3.18	3.07	2.58	2.39	2.12	2.60	2.12	2.53
0.20	4.23	4.08	4.04	3.32	2.93	2.69	2.39	2.66	3.34
0.25	6.24	5.25	5.16	3.70	2.81	3.28	3.31	3.14	3.67
0.30	9.57	7.24	6.57	4.92	3.48	4.35	3.69	3.52	4.72
0.35	15.86	10.98	9.20	6.21	4.02	4.40	4.55	3.95	4.94
0.40	27.58	18.60	14.16	8.69	5.61	5.24	4.59	4.73	5.28
0.45	50.23	36.53	23.79	13.85	7.53	6.28	5.40	4.92	5.76
N _G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	5.31	4.58	3.85	3.64	3.11	2.85	2.60	2.29	2.17
0.10	6.82	5.73	4.79	4.42	3.68	3.38	2.90	2.51	2.32
0.15	8.57	6.95	5.78	5.28	4.68	4.01	3.40	3.15	2.65
0.20	10.41	8.62	7.10	6.19	5.48	4.60	4.14	3.60	2.93
0.25	12.87	10.63	8.40	7.36	6.76	5.28	4.53	4.02	3.27
0.30	15.00	12.68	10.22	8.85	7.96	6.34	5.41	4.61	3.72
0.35	16.31	14.58	11.88	10.63	9.46	7.42	6.22	5.33	4.18
0.40	17.00	15.78	13.35	12.50	11.27	9.00	7.30	6.01	4.69
0.45	18.13	15.72	14.58	14.22	13.37	10.65	8.64	7.09	5.37
N _B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	6.13	5.26	4.34	4.05	3.49	3.15	2.75	2.30	2.14
0.10	7.66	6.50	5.45	4.86	3.93	3.61	3.05	2.47	2.25
0.15	9.64	7.87	6.42	5.65	4.85	4.13	3.45	2.95	2.47
0.20	12.07	9.78	8.02	6.75	5.78	4.72	4.09	3.36	2.68
0.25	15.11	12.24	9.85	7.97	6.81	5.34	4.55	3.70	3.00
0.30	18.52	15.07	12.26	9.73	8.04	6.33	5.26	4.21	3.35
0.35	21.19	18.52	14.85	12.17	9.62	7.36	6.15	4.80	3.76
0.40	22.43	21.01	17.95	14.78	11.82	9.07	7.08	5.52	4.22
0.45	21.13	20.95	20.26	18.09	14.48	10.94	8.49	6.58	4.83

Table 5.10 Adaptive SHT, $320 \leq N_G \leq 480$ & $80 \leq N_B \leq 120$ (40~60%)

FP (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	2.48	3.04	3.30	2.96	2.03	2.75	2.63	1.26	1.35
0.10	2.65	2.82	2.22	3.06	2.20	2.52	2.81	1.48	1.90
0.15	2.66	2.92	2.66	2.92	2.32	2.06	2.85	1.65	1.68
0.20	2.78	2.79	2.80	2.55	2.79	2.14	3.23	1.69	1.14
0.25	3.02	3.00	3.03	2.79	2.80	2.59	3.45	2.08	1.15
0.30	3.48	3.28	3.06	2.88	2.90	2.32	3.24	2.28	1.22
0.35	3.71	3.48	3.26	3.05	2.66	2.79	2.70	2.36	1.38
0.40	4.04	3.54	3.24	2.81	2.52	2.90	2.93	2.59	1.74
0.45	4.57	3.88	3.36	2.90	2.80	2.98	2.96	2.65	1.91
FN (%)	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	2.79	1.95	1.90	1.92	1.80	1.32	1.20	1.71	1.23
0.10	3.96	3.75	3.33	2.62	3.00	2.33	1.93	1.76	1.62
0.15	4.57	3.87	4.72	3.33	4.34	3.27	2.48	2.69	2.55
0.20	4.72	4.35	4.35	3.29	3.10	3.38	2.41	3.11	3.28
0.25	4.92	4.77	4.35	4.08	3.09	3.80	2.77	2.83	3.60
0.30	5.50	5.24	4.66	4.09	3.57	3.87	3.06	3.23	3.54
0.35	6.88	5.71	5.13	5.02	4.23	3.94	3.56	3.48	3.93
0.40	8.77	6.53	5.95	5.18	4.43	4.51	3.67	3.60	3.87
0.45	26.57	11.39	6.99	5.90	4.96	3.98	3.80	4.12	3.80
N _G	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	4.20	3.79	3.41	3.10	2.82	2.73	2.48	2.19	2.08
0.10	5.61	4.69	4.12	3.78	3.27	3.04	2.75	2.48	2.29
0.15	7.69	6.46	5.18	4.72	3.79	3.54	3.26	2.78	2.47
0.20	10.64	8.45	7.00	6.25	5.29	4.46	4.19	3.36	2.84
0.25	13.75	11.04	9.05	7.59	6.75	5.27	4.74	4.04	3.20
0.30	17.14	14.30	11.51	9.62	8.16	6.32	5.69	4.60	3.66
0.35	21.17	17.74	14.60	11.47	9.93	7.95	6.58	5.38	4.28
0.40	25.23	21.62	17.92	14.75	12.28	9.64	7.90	6.35	4.99
0.45	24.20	24.77	21.89	18.10	14.77	12.20	9.82	7.55	5.99
N _B	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	6.46	5.14	4.29	3.74	3.50	2.92	2.33	2.23	2.02
0.10	8.16	6.51	5.75	4.51	4.05	3.40	2.64	2.45	2.03
0.15	10.34	8.16	6.77	5.50	4.71	4.11	2.99	2.71	2.22
0.20	13.16	10.36	8.26	6.90	5.33	4.70	3.36	3.04	2.55
0.25	16.04	13.02	10.09	8.28	6.60	5.26	3.75	3.30	2.78
0.30	19.10	15.91	12.66	10.11	7.98	6.40	4.51	3.71	3.06
0.35	22.81	19.26	15.59	12.49	9.92	7.40	5.61	4.20	3.36
0.40	26.48	22.86	18.82	15.74	12.56	9.01	6.60	4.88	3.66
0.45	25.77	25.77	22.55	18.89	15.07	11.27	8.14	5.75	4.08

Table 5.11 Adaptive SHT, $N_G : N_B$ (40~60%)

THETA0	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.561	0.593	0.641	0.701	0.754	0.827	0.857	0.912	0.957
0.10	0.536	0.591	0.631	0.692	0.745	0.796	0.874	0.910	0.961
0.15	0.545	0.604	0.629	0.715	0.754	0.809	0.868	0.938	0.958
0.20	0.545	0.608	0.640	0.682	0.748	0.810	0.855	0.917	0.947
0.25	0.545	0.576	0.657	0.697	0.741	0.801	0.872	0.913	0.965
0.30	0.543	0.606	0.645	0.682	0.741	0.811	0.866	0.918	0.963
0.35	0.563	0.590	0.655	0.672	0.742	0.803	0.850	0.925	0.967
0.40	0.573	0.624	0.655	0.698	0.740	0.796	0.842	0.901	0.961
0.45	0.569	0.624	0.644	0.692	0.747	0.798	0.858	0.915	0.957

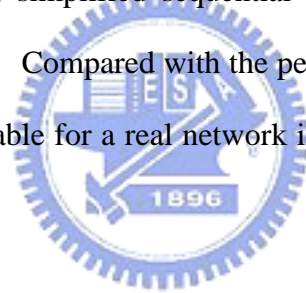
THETA1	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.05	0.045	0.041	0.039	0.049	0.043	0.040	0.040	0.043	0.052
0.10	0.090	0.086	0.081	0.079	0.080	0.082	0.094	0.072	0.106
0.15	0.130	0.136	0.137	0.155	0.138	0.137	0.141	0.148	0.146
0.20	0.202	0.206	0.184	0.205	0.192	0.209	0.203	0.225	0.183
0.25	0.240	0.239	0.242	0.234	0.230	0.254	0.246	0.246	0.255
0.30	0.264	0.285	0.307	0.295	0.274	0.271	0.308	0.293	0.315
0.35	0.323	0.323	0.345	0.336	0.357	0.345	0.360	0.360	0.356
0.40	0.347	0.367	0.380	0.381	0.381	0.418	0.407	0.397	0.391
0.45	0.379	0.399	0.404	0.430	0.417	0.445	0.455	0.434	0.455

Table 5.12 θ_0 and θ_1 of Adaptive SHT, $N_G : N_B$ (40~60%)

Chapter 6

Conclusion

We have investigated three existing schemes for detecting scanning worms – sequential hypothesis testing, simplified sequential hypothesis testing, and reverse sequential hypothesis testing. Compared with the performance of these schemes, we think that they may be unsuitable for a real network if some parameters are unknown and estimated erroneously.



We have presented in this paper an adaptive sequential hypothesis testing scheme for fast detection of scanning worms. The adaptive estimation procedure can adjust θ_0 and θ_1 automatically according to the information collected previously and make the sequential hypothesis testing algorithm more robust to variation of θ_0 and θ_1 . The proposed adaptive detection algorithm provides accurate estimates of θ_0 and θ_1 and thus achieves false positive and false negative probabilities close to the desired values.

Bibliography

- [1] A. Wald. *Sequential Analysis*. J. Wiley & Sons, New York, 1947.
- [2] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. “Fast Portscan Detection Using Sequential Hypothesis Testing.” In *Proceedings of the IEEE Symposium on Security and Privacy*, May 9-12 2004.
- [3] S. E. Schechter, J. Jung , and A. W. Berger. “Fast Detection of Scanning Worms Infections.” In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*, September 15-17 2004.
- [4] N. Weaver, S. E. Schechter, V. Paxson. “Very Fast Containment of Scanning Worms.” In *Proceedings of the 13th USENIX Security Symposium*, August 9-13 2004.
- [5] N. Weaver, V. Paxson, S. Staniford ,and R. Cunningham. “A Taxonomy of computer worms.” In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, pages 11–18. ACM Press, October 27, 2003.
- [6] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. “Inside the Slammer Worm.” In *Proceedings of the IEEE Symposium on Security and Privacy*, 1:33–39, July 2003.
- [7] S. Staniford, V. Paxson, and N. Weaver. “How to Own the Internet in your spare time.” In *Proceedings of the 11th USENIX Security Symposium*, August 7–9, 2002.
- [8] J. Twycross and M. M. Williamson. “Implementing and testing a virus throttle.” In *Proceedings of the 12th USENIX Security Symposium*, August 4–8 2003.
- [9] Type I and Type II errors. From Wikipedia, the free encyclopedia.

http://en.wikipedia.org/wiki/Type_I_and_type_II_errors

- [10] C. C. Zou, D. Towsley, W. Gong, and S. Cai. “Routing Worms: A Fast, Selective Attack Worm based on IP Address Information.” In *Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation (PADS’05)*, June 2005.
- [11] D. Moore, C. Shannon, and J. Brown. “Code-Red: a Case Study on the Spread and Victims of an Internet Worm.” In *Proceedings of ACM/USENIX Internet Measurement Workshop*, France, November 2002.
- [12] CAIDA. Dynamic graphs of the Nimda worm,
<http://www.caida.org/dynamic/analysis/security/nimda>.
- [13] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber. “A Network Security Monitor.” In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 296–304, 1990.
- [14] M. Roesch. “Snort: Lightweight Intrusion Detection for Networks.” In *Proceedings of the 13th Conference on Systems Administration (LISA-99)*, pages 229–238, Berkeley, CA, Nov. 7–12 1999. USENIX Association.

簡歷

一、個人資料

姓名	林建成
生日	1983/01/12
出生地	台灣彰化
E-mail	aircheng.cm94g@nctu.edu.tw
永久地址	202 基隆市中正區平一路 23 巷 4 號 3 樓
通訊地址	300 國立交通大學電信工程學系 823 實驗室
電話	(03) 571-2121 轉 54570
現職	國立交通大學電信工程學系 系統組 碩士班二年級

二、學歷

國小	基隆市立和平國民小學	1989/09 – 1995/06
國中	基隆市立中正國民中學	1995/09 – 1998/06
高中	台北市立建國高級中學	1998/09 – 2001/06
大學	國立交通大學 電信工程學系	2001/09 – 2005/06
研究所	國立交通大學 電信工程學系碩士班	2005/09 – 2007/07