View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by UUM Repository

A Traffic Signature-based Algorithm for Detecting Scanning Internet Worms

Mohammad M. Rasheed, Osman Ghazali, Norita Md Norwawi and Mohammed M. Kadhum

Graduate Department of Computer Science, College of Arts and Sciences,
Universiti Utara Malaysia
06010 UUM Sintok, MALAYSIA
E-mail: mohmadmhr@yahoo.com E-mail: {osman, nmn, kadhum } @uum.edu.my

Abstract: Internet worms that spread autonomously from one host to another cause major problem in today's networks. On 25th January 2003, "Slammer" was released into the internet and after ten minutes the worm infected more than 90% of vulnerable hosts. Worms cause damage to the network by consuming its resources such as bandwidth. In this paper, we propose a method for detecting traffic signature for unknown internet worm. The proposed method has two algorithms. The first part is an Intelligent Failure Connection Algorithm (IFCA) using Artificial Immune System; IFCA is concerned with detecting the internet worm and stealthy worm. In order to reduce the number of false alarm, the impact of normal network activities is involved but TCP failure and ICMP unreachable connection on same IP address are not calculated because the internet worm strategic attack on the different IP address. The second algorithm Traffic Signature Algorithm (TSA) is concerned with capturing traffic signature of the scanning internet worm. In this paper, we show that the proposed method can detect traffic signature for MSBlaster worm.

Keywords: Internet worm Detection, Firewall, Generate Signatures, Router.

1. Introduction

Worms are widely regarded to be a major security threat facing the Internet today. Active worms spread in an automated fashion, which can flood the Internet in a very short time. Incidents such as Slammer worm that infected more than 90% of vulnerable machines within ten minutes on January 25th, 2003 [1] is the example of worms' threats. Therefore, worm attacks present significant threats to the Internet. Flash Worms can attack with high speed of spreading, but stealth worms spread much slower that makes detection hard [11].

Anti-virus is the popular tool to combat worms. It used signature based technology [2] to detect worms. However the high spreading speed of worm results in anti-virus is less effective in detecting worms. Moreover, anti-virus cannot detect unknown internet worm automatically because it uses signature in detecting worms. Anti-virus compares the file structure of the worms with the signatures stored in its database. If they are matched, then the file is considered as has been infected by the worm. This required the anti-virus database to be frequently updated, so that it can detect new worms. This is the main reason why anti-virus cannot detect most of unknown internet worm automatically. Beside anti-virus, firewalls and routers can be used to detect worm signature and block the worm, but this occurs only after the worm already spread. The worm generates an IP address and uses that IP address to communicate to potential victim, when the IP address is unused; the router returned an ICMP "Destination Unreachable" to infector computer [3] (see figure 1).

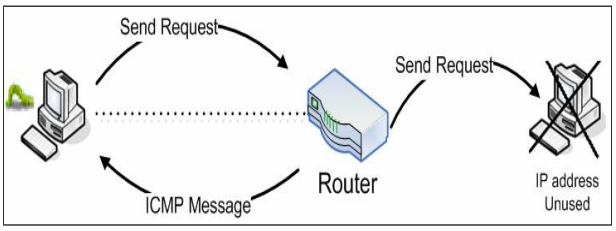

Figure 1. ICMP message

When the worm sent a SYN packet to a used IP address with destination port closed, TCP RESET packet is returned [3] (see figure 2).

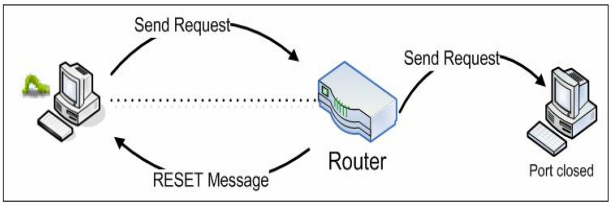

Figure 2. RESET message

The technology of internet worm detection is to check the way of the control message, such as ICMP destination unreachable message and RESET in TCP.

In this work, we propose the AIS to compute a threshold that can help in detecting the Internet worm. In addition, we propose an intelligent way to compute the threshold range for detecting new types of worms. The False alarm of our proposed is reduced. Overall, the proposed of IFCA algorithm is concerned with detecting the rapid internet worm and stealthy internet worm, while TSA is concerned with capturing traffic signature of the internet worm.

Schechter et al. [4] proposed worm detection method based on the failed connection. This algorithm can detect internets worm but does not work well on detecting stealthy worm. The threshold used in that algorithm cannot detect stealthy worm.

Chen & Tang [5] analyzed the essential character of TCP-based worm's propagation that sending out a large number of TCP connection requests. They proposed an effective approach to detect network worms based on the number of failure connection received by the network routers. The

approach can be divided into two phrases: short term and longer term. This strategy may works well on detecting randomly scanning worm and stealthy worm. However, the impact on normal network activities has not been considered. In addition, the rate of false alarms could be large and take long time to detect the worm.

Yang et al. [6] proposed a worm detection algorithm that has two sub algorithms. The first sub algorithm, the "short term algorithm", runs well to detect internet worm. While the second sub algorithm, the "longer term algorithm", cannot detect all types of the stealthy worm (see Table 1).

Algorithm Name	Rapid Worm Detection	Stealthy worm Detection	Signature Detection	Speed
Schechter [4]	(√)	-	-	Slow
Chen [5]	(√)	(√)	-	Slow
Yang [6]	(√)	(√) but some w cannot detect it	-	fast
Our proposed algorithm	(√)	(√)	(√)	Faster than Yang's algorithm

Table I. Mechanisms Analysis

The focus of this paper is on TCP-based worms. The remainder of this paper is organized as follows. Section 2 describes the design and Experiments of intelligent failure connection algorithm. Section 3 describes the design of traffic signature algorithm. Section 4 discusses the Experiment result. Finally Section 5 concludes this paper.

2. IFCA

In this section, we present IFCA that based on Artificial Immune System; the Artificial Immune System recognize between self and non-self. An Artificial Immune System (AIS) is a bio-inspired classification system which is derived from the Human Immune System (HIS). AIS are one of the most recent approaches in computational intelligence. They provide efficient information processing capabilities [7]. IFCA appoints the difference between regular connection and worm connection. The worm scans different IP addresses every second. IFCA depends on the TCP failure connection and ICMP unreachable message on different random addresses. Therefore, there will be a large number of failure connections if the computer has infected by a worm.

2.1 Design of IFCA

IFCA mechanism records the number of failed connection packets such as ICMP and TCP RESET packets that are returned from the external destination address to the internal forged. It monitors source IP address placed in the router (see figure 3). Once detecting the first failed connection packets, the algorithm then extracts the source address, source port, destination address, and destination port from the packet and creates the record. The false positive rate is largely reduced when IFCA received normal connection, i.e. TCP SYN/ACK, "counter" will be decreased. Also, IFCA ignores the packet when the destination IP is recorded into the counter table because the internet worm attack strategy is "attacking different IP address".

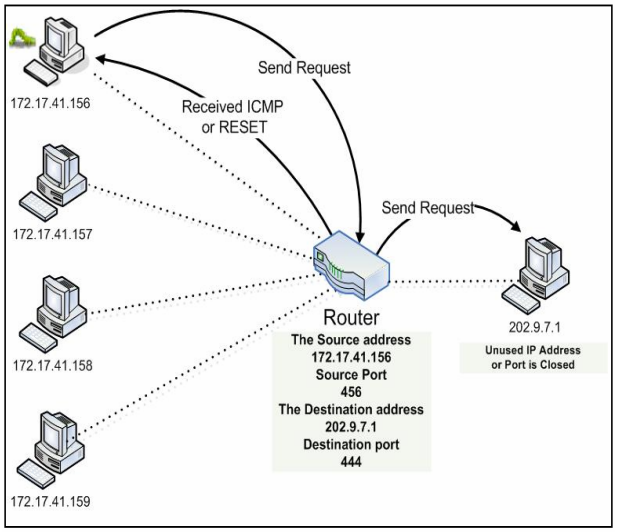

Figure 3. Error Message Returned to Router

Only the first failed connection sent from the forged source IP address to different destination IP address is recorded. IFCA will remove the "counter" every three days.

β =100/minute failed rate of threshold. Then *X= (1 to n)* average of failure connection (AFC) in one minute.

AFC = Counter/Minute *(1)*

IFCA can calculate the average of failure connection after five second when IFCA received first failure connection.

Threshold can be processed by the following equation of Summation of threshold (ST):-

ST = 2^ (6.65 + 0.050054 (β -X)) *(2)*

Yang's algorithm provides one threshold in long term algorithm therefore the process in this algorithm need long time for detecting stealth worm.

The Yang's algorithm [6] detects the internet worm if the failure connection is less than 100/minute failure connections by using "short term algorithm". When the failure connection is equal or greater than 3000/day failure connection the Yang's algorithm detects this type of stealthy internet worm by using "long term algorithm". Our algorithm uses same Yang's algorithm warning.

Example (1) when we use Yang's algorithm to detect the internet worm that has 3000/day failure connections, we can calculate the average number of failure connections when we use IFCA to detect same worm properties. The average number of failure connections will be:

➔ 3000/1440 (one day= 1440 minute)

➔2.08333/minute (average of failure connections).

Then ST will be:

ST= 2^ (6.65 + 0.050054 (β -X))

➔ ST= 2^ (6.65 + 0.050054 (100 -X))

➔ ST= 2^ (6.65 + 0.050054 (100 -2.08333))

➔ ST≅3000 IFCA detects the worm after one day.

IFCA and Yang algorithm are detecting the worm after one day (1440 minute).

Example (2) when we use Yang's algorithm to detect the internet worm that has 100/minute failure connections, in this case we can calculate the average of failure connections,

when we use IFCA to detect same worm properties and average of failure connection will be:

➔ 100/1

➔ 100/minute (average of failure connections).

Then ST will be:

ST= 2^ (6.65 + 0.050054 (β -X))

➔ ST= 2^ (6.65 + 0.050054 (100 -X))

➔ ST= 2^ (6.65 + 0.050054 (100 -100))

➔ ST≅100 IFCA detects the worm on one minute.

IFCA and Yang algorithm are detecting the worm on one minute.

Our algorithm uses different threshold values over different time periods; therefore our method is faster than Yang's Algorithm when the worm is less than 3000/day or less 100/minute failure connections. Also, IFCA detects the worm when that has greater than 3000/day failure connections, unlike Yang's algorithm.

The IFCA equation depends on the average of failure connection to compute the threshold. IFCA can detect the worm early in usual time. But if the worm cannot be detected n early stage, the algorithm provides more time and new threshold to detect the worm.

IFCA can detect the worm by calculating different time on different number of failure connections.

ST should be greater than fifteen. Else the traffic will be forwarded.

T1= (ST / average of failure connection) (3)

T2= (time now – time start of the algorithm) (4)

Unlike Yang's algorithm, IFCA is more dynamic in detecting the worm because it calculates the threshold every time. IFCA detects the worm by compare T1 to T2 as follows: If (T2 is small or equal to T1) and (the counter is greater than or equal to the summation of Threshold) the worm is detected. Else check T1, T2. If (T2 is greater than T1), then go to feed back and decrease the average with new calculation to give another chance to detect the worm. If T1 small than T2, then the traffic will be forwarded because it is a normal connection. Whenever the counter value does not exceed the threshold during time cumulative computation phase, the traffic sent from the corresponding IP address would be forwarded as normal activity (see figure 4).

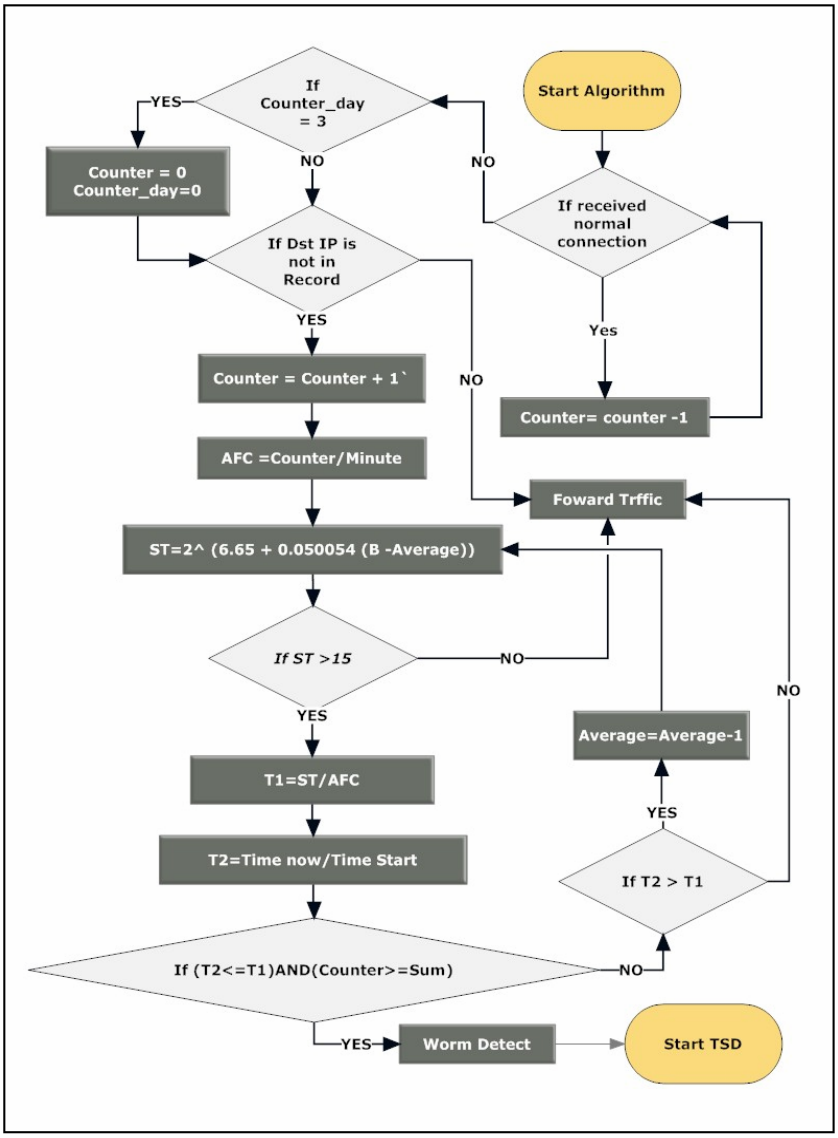

Figure 4. The flow chart of the IFCA

2.2 Experiments of IFCA

In this section, we describe the simulation experiments of IFCA to measure the effectiveness of detecting the rapid scanning worm and stealthy worm.

Section 2.2.1 shows results for detecting rapid internet worm, section 2.2.2 shows results for detecting stealthy internet worm; section 2.2.3 shows the detection of other types of stealthy internet worms. The result in sections 2.2.1, 2.2.2 and 2.2.3 are obtained by using the same types of worms that are tested on the two different algorithms namely Yang algorithm Yang et al. [5] and IFCA, and The key parameters used by IFCA and Yang algorithm are set as follows:

N=36000 the total number of vulnerable hosts
I (0) =1number of initial infected hosts
β =100/minute failed rate of threshold
Rapid Scanning Worm rate =120/min, 150/min
Stealthy Scanning Worm rate = 93 /min, 88/ min, 2360/day

2.2.1 Detecting Rapid Scanning Worm

Figures 5 and 6 show two types of worms that are detected using Yang et al. [6] algorithm. Figure5 shows the average of failure connections which is 120/minute, and the processing time to detect a worm is 50 sec. Figure 6 shows the average of failure connections which is 150/minute, and the processing time to detect the worm is 40 sec.

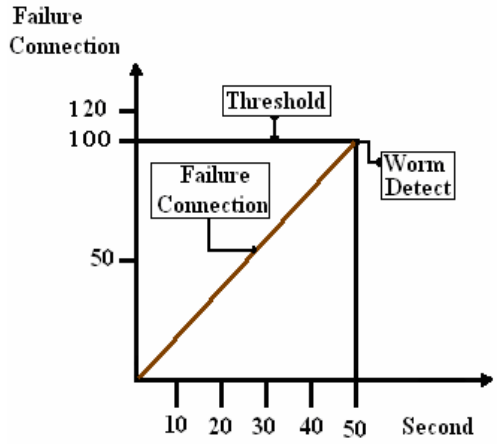

Figure 5. Yang algorithm detected the worm after 50 sec

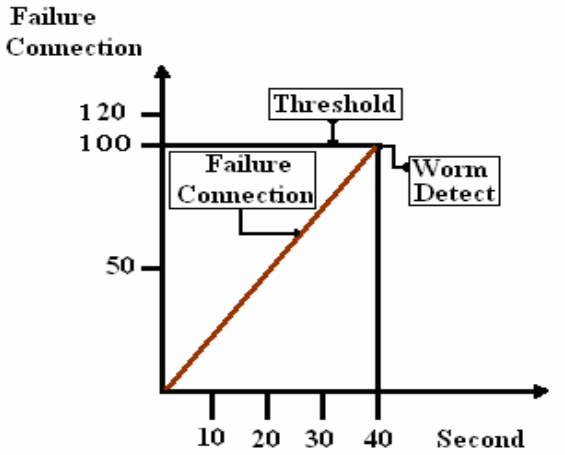

Figure 6. Yang algorithm detected the worm after 40 sec

Figures 7 and 8 show two types of worms that are detected using IFCA. Figure 7 shows the average of failure connections which is 120/minute, and the processing time to detect the worm is 25 sec. In figure 8, the average of failure connections is 150/minute, and the processing time to detect the worm is 7 sec.

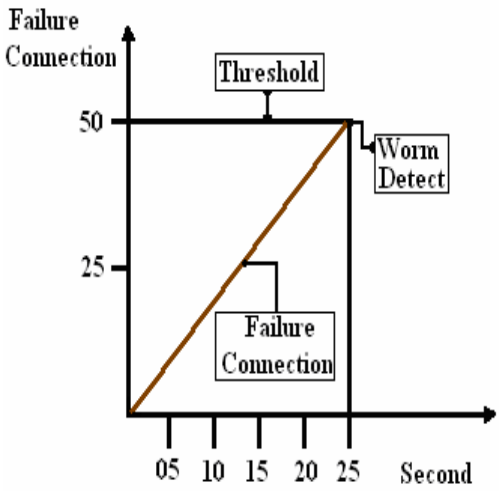

Figure 7. IFCA detected the worm after 25 sec

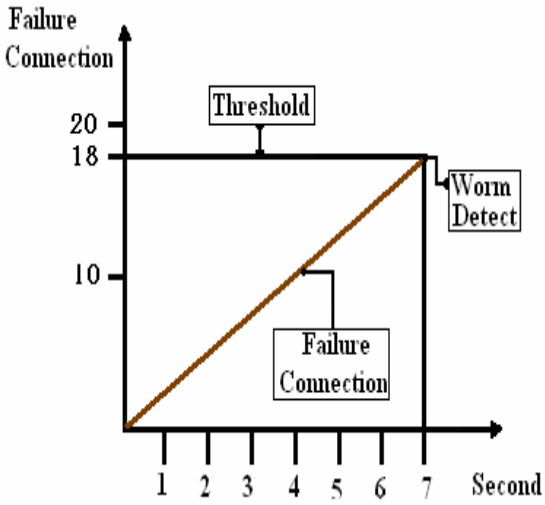

Figure 8. IFCA detected the worm after 7 sec

It is noticeable that IFCA operates faster than Yang et al. [6] algorithm in detecting rapid internet worm.

2.2.2 Detecting Stealthy Internet Worm

Figures 9 and 10 show two types of worms that are detected using Yang et al. [6] algorithm. Figure 9 shows the average of failure connection which is 88/minute, and the processing time to detect a worm is 34min and 5sec Figure 10 shows the average of failure connection which is 93/minute, and the processing time to detect the worm is 32min and 15sec.

Figures 11 and 12 show two types of worms that are detected by IFCA. Figure 11 shows the average of failure connection which is 88/minute, and the processing time to detect the worm is 103 sec. In figure 12, the average of failure

connection is 93/minute, and the processing time to detect the worm is 82sec.

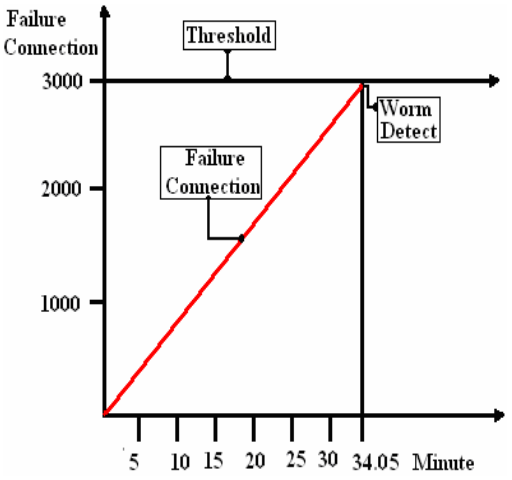

Figure 9. Yang algorithm detected the worm after 34min 5 sec

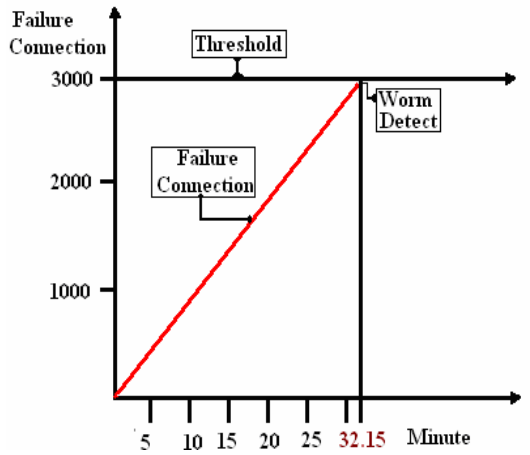

Figure 10. Yang algorithm detected the worm after 32 min 15 sec

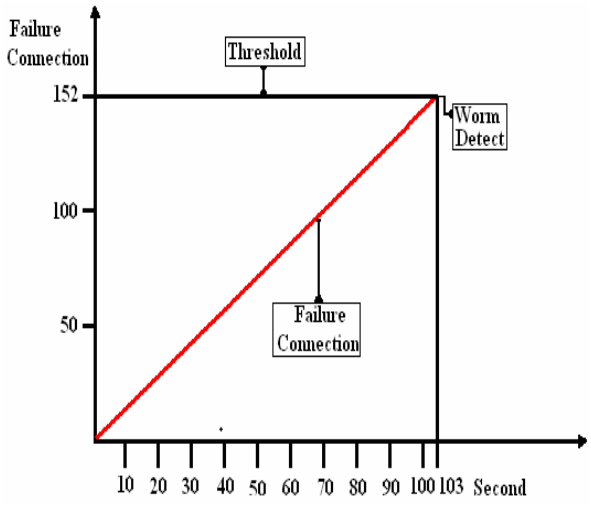

Figure 11. IFCA detected the worm after 103 sec

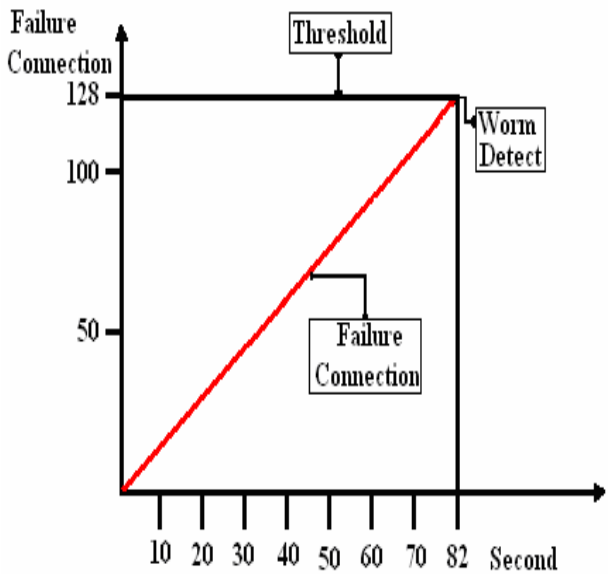

Figure 12. IFCA detected the worm after 82 sec

This is proof that IFCA operates faster than Yang et al. [6] algorithm in detecting stealthy internet worm.

2.2.3 Detecting another Stealthy Internet Worm

In this section, we used Yang algorithm to detect a worm. The worm has failure connection of 2360/day and the result after 30 hours is that the algorithm cannot detect this worm as shown in figure 13. Yang algorithm's [6] has to check again the system after 24 hours. Yang algorithms cannot detect this type of worm. This worm has properties less than 3000/day failure connection.

In the second experiment, we used the IFCA to detect the worm. The worm has failure connection of 2360/day, and after 30 hours IFCA can detect this worm as shown figure 14.

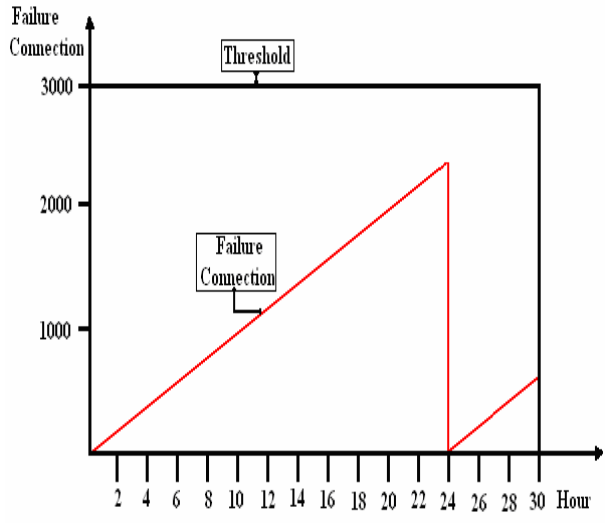

Figure 13. Yang algorithm cannot detected the worm after 30 hours

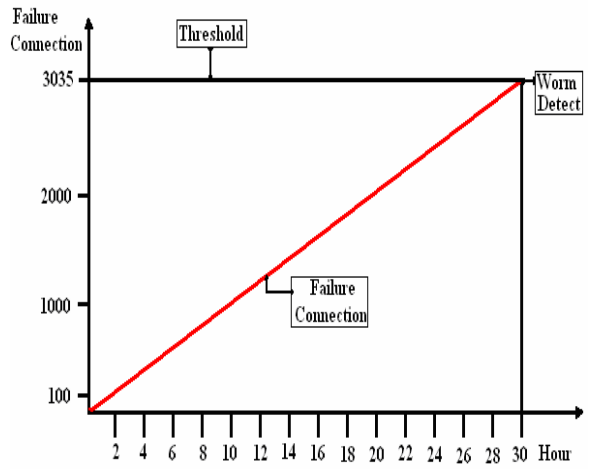

Figure 14. IFCA detected the worm after 30 hours

The results show that IFCA detects other types of worms, while Yang algorithm fails to detect these types of worms.

3. TSA

TSA mechanism detects traffic unknown internet worm depending on source IP address number that was returned by router so that we can collect the packet by using packets monitor. The mechanism depends on captures all the packets synchronization with successful replica from the infector to the victim

TSA works when the worm is detected by IFCA, then internet worm signature can be detected by using traffic signature monitor.

The different infection sequences might have different ports. For example, in the MSBlaster worm, the source ports vary with different infection sessions, which means the source ports can be changed, while the destination ports are fixed [9]. In this case, our mechanism uses the destination port for packet capturing. But other worms may have different strategy; the source port is fixed like Witty worm but destination port is changed [10]. In this case, our method uses the source port for packet capturing. The packet capturing means capture all packets between infector and victim when the port was opened at the victim side during sending request by the infector computer.

The algorithm on focuses successful traffic synchronization and captures all these packets by the traffic signature monitor. In order to reduce the number of false alarm, the algorithm will check whether the destination IP is in the record. If the destination IP is in the record that means that the current connection is normal because the worm generates different IP addresses. If the traffic synchronization is not successful, the worm searches for other servers to infect it.

The algorithm of TSA compares the packe
ts and takes the successful synchronization of the packets that are similar in traffic synchronization, (See figure 15).

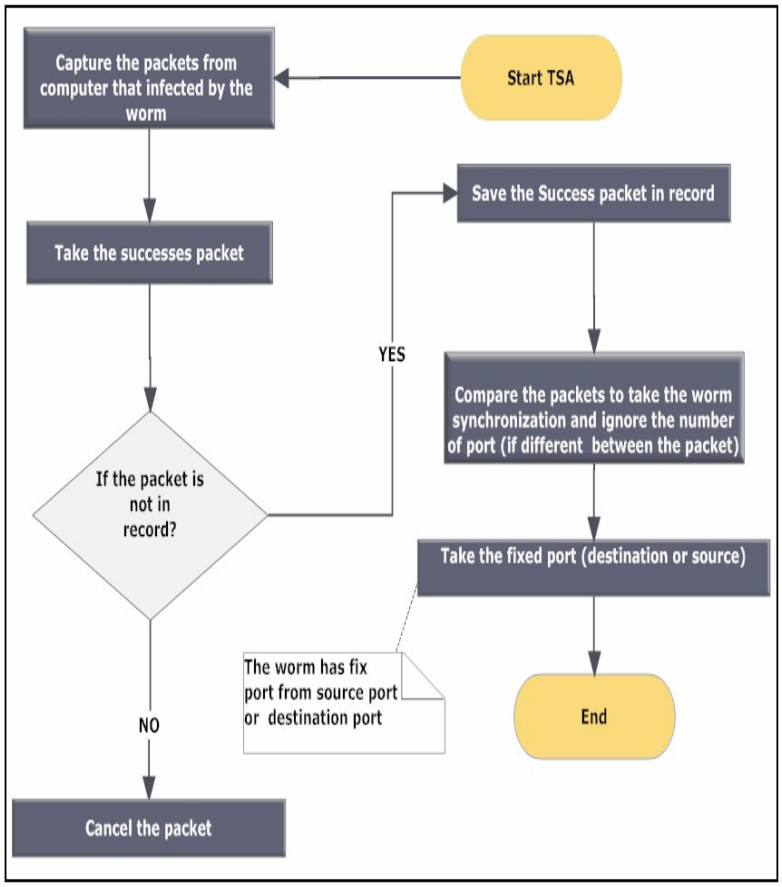

Figure 15. The flow chart of the TSD

The worm has a successful connection (see figure 16) where computer B is accepted the connection that includes the worm from computer A, because the worm generates random IP address in computer A and the same IP was used to connect to computer B. The worm will be transferred from computer A to computer B when the port in computer B is open.

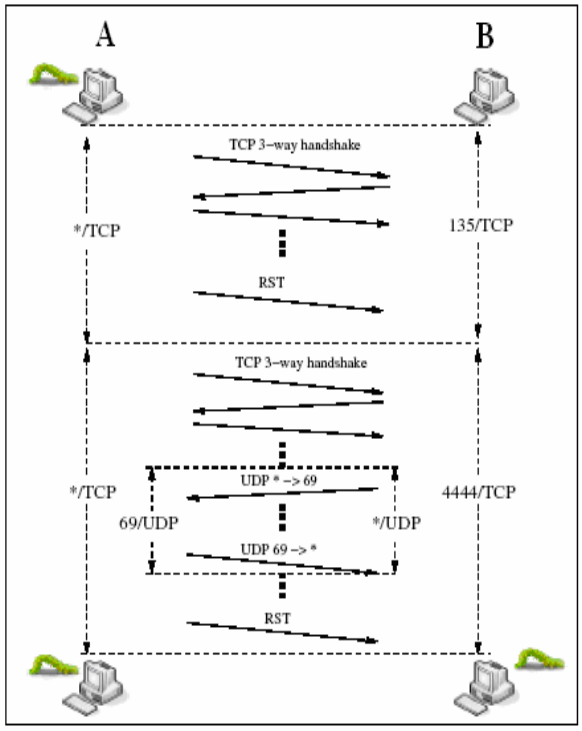

Figure 16. Sequence of Infected Worm [8]

4. Experiment Result

This section show the experiment synchronization of worm was successful through TSA. When the worm is detected by IFCA, then internet worm signature can be detected by using TSA. The algorithm captures all the packets synchronization with successful replica from the infecter to the victim. TSA started from destination port for MSBlaster worm because the destination port is fix port while source port is not fix port.

The traffic signature for MSBlaster worm is detected as follow.

< TCP, X1 /infecter, 135/victim, SY N >

< TCP, 135/victim, X1/infecter, SY N, ACK >

< TCP, X1/infecter, 135/victim, ACK >

< TCP, X1/infecter, 135/victim, RST >

< TCP, X2/infecter, 4444/victim, SY N >

< TCP, 4444/victim, X2/infecter, SY N, ACK >

< TCP, X2/infecter, 4444/victim, ACK >

< UDP, X3/victim, 69/infecter >

< UDP, 69/infecter, X3/victim >

< TCP, X2/infecter, 4444/victim, RST >

Source port for X1, X2, and X3 are not fix port, while destination port is 135,4444,69 there are fix port.

5. Conclusions

The worm is very fast spread and the current techniques to detect the internet worms are slow. This paper presents a new method for detecting the worm and generating the traffic signature automatically. The results of the experiments show that the algorithm detected the traffic signature for MSBlaster worm. Also, the proposed algorithm can detect stealthy worm.

References

[1] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer Worm", *IEEE Security and Privacy*, vol. 1, no. 4, pages 33–39, Jul. 2003.

[2] T. Alagna., *Defending the Digital You: How to Fight Online Identity Theft, America*, Larstan, 2005.

[3] [3] D. Ellis, J. Aiken, K. Attwood, and S. Tenaglia. "A Behavioral Approach to Worm Detection", *Proceedings of the Second ACM Workshop on Rapid Malcode (WORM)*, pages 43 – 53, Oct. 2004.

[4] S. Schechter, J. Jung, & A. Berger. "Fast Detection of Scanning Worm Infections", *Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID)*, Sophia Antipolois, France, Sep 2004.

[5] S. Chen & Y. Tang. "DAW: A Distributed Antiworm System", *IEEE Journal*, Volume 18, Issue 7, Pages 893 – 906, Jan 2007.

[6] X. Yang, J. Lu, Y. Zhu & P. Wang. "Simulation and Evaluation of a New Algorithm of Worm Detection and Containment", *Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*, Taiwan, pages 448-453, Dec 2006.

[7] S. Schaust & M. Drozda . "Influence of Network Payload and Traffic Modelson the Detection Performance of AIS", *IEEE International Conference*, pages 44-51, 2008.

[8] X. Jiang and D. Xu, "Profiling Self-Propagating Worms via Behavioral Footprinting", *Proceedings of ACM Workshop on Recurring Malcode*, Nov. 2006.

[9] MSBlaster Worms. http://www.cert.org/advisories/CA-2003-20.html, 2003.

[10] Witty Worm. http://www.us-cert.gov/current/archive/2004/03/22/archive.htm l#witty , 2004.

[11] S. Staniford, V. Paxson, and N. Weaver. "How to Own the Internet in Your Spare Time". *In Proceedings of the USENIX Security Symposium*, pages 149–167, 2002.