http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (7): 760-767, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.760.767

Intelligent Signature Detection for Scanning Internet Worms

[1,3]Mohammad M. Rasheed, [1]Osman Ghazali and [2]Norita Md Norwawi
[1]Department of Graduate Computer Science, College of Arts and Sciences,
Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia
[2]Faculty of Science and Technology, Universiti Sains Islam Malaysia,
71800 Nilai, N. Sembilan, Malaysia
[3]Telecommunication Research Center, Information Technology Directorate,
Ministry of Science and Technology, Iraq

Abstract: Worms are widely regarded to be a major security threat faced by the Internet today. Active worms spread in an automated fashion, which can flood the Internet in a very short time, Incidents such as Conficker worm, detected in November 2008, a computer worm targeting the Microsoft Windows operating system, once infected 15 million hosts. In this study, we detect DNA signature by scanning internet worm using three algorithms. The first part, Intelligent Failure Connection Algorithm (IFCA) by using Artificial Immune System, is concerned with detecting the internet worm and stealthy worm in which computer infected by the worm. The second part, Traffic Signature Algorithm (TSA), is concerned with capturing traffic signature for the worm from infector computer. Finally, the third stage is DNA Converter Signature (DNACS) which converts traffic signature to DNA signature and sends it to DNA Filtering. In this study, we show that our proposed technique can detect DNA signature for MSBlaster worm.

Key words: Internet worm detection, firewall, generate signatures, router

INTRODUCTION

The Witty is a type of internet worm appeared in 2004 infected 110 hosts in the first 10 seconds and 160 at the end of 30 sec. In a November 2008, Conficker worm spread on the internet and targeting is the Microsoft Windows operating system, once infected fifteen million hosts (Dengyin and Ye, 2010).

The defense from unknown malware are challenging tasks (Rozenberg *et al*., 2008). Automatic worm detection is challenging task because it is difficult to expect what form that the next worm will take (Costa *et al*., 2005).

Furthermore, the internet worm has high spreading speed of worm consequently made the current technique of anti-virus less effective in detecting worms. Worm detection technique are divided into a signature-based technique and Anomaly detection technique (Li *et al*., 2008). Anti-virus systems and most current intrusion detection systems are signature-based technology (Min and Gupta, 2009; Mohammed *et al*., 2010; Moskovitch *et al*., 2009; Zolkipli and Jantan, 2010), the problem in signature-based technology is that they can only detect known attacks with identified signatures that are produced recently (Tang and Chen, 2005). Beside anti-virus, firewalls and routers can be used to detect worm signature and block the worm (Muda *et al*., 2011; Yu *et al*., 2009) but this reactive response happens only after the worm already spread. The detection system must therefore be able to handle known as well as unknown threats (Nasir *et al*., 2008). Yoo *et al*. (2001) enhance the detection system to detect unknown attack only but our detection is different because detect the unknown signature.

The internet scanning worm generates the IP address whenever the IP address is unused; the router returned the Internet Control Message Protocol (ICMP) "Destination unreachable" to the infected computer (Fig. 1).

When the worm sent a TCP/SYN packet to a used IP address with destination port closed, TCP RESET/ACK the infected computer will receive it (Fig. 2).

Schechter *et al*. (2004) have proposed worm detection method based on failed connections. This is a hybrid approach combining sequential hypothesis testing and connection rate limiting. This method detects infected hosts using a small fraction of network events. The algorithm named reverse sequential hypothesis test reduces the number of first contact connections required

Corresponding Author: Mohammad M. Rasheed, Department of Graduate Computer Science, College of Arts and Sciences, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

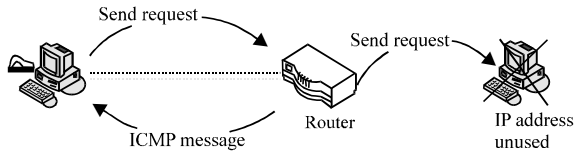

Fig. 1: ICMP message

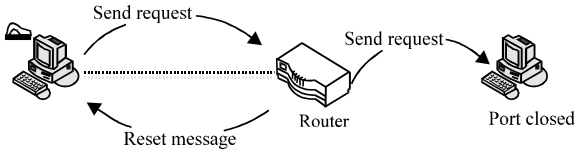

Fig. 2: RESET message

to be observed for the detection of scanning. In addition, the number of observations required by this algorithm to detect hosts' scanning behavior is not affected by the presence of benign network activity. The other algorithm known as the new credit-based algorithm limits the rate at which a host may issue the first-contact connections that are indicative of scanning activity. This algorithm reduces the false positive alarms significantly. As these two algorithms are complementary, combining these approaches results in an improved worm detection probability. The main shortcoming of this method is that it does not work well in detecting stealthy worms.

Yang *et al*. (2006) have proposed a worm detection algorithm called Improved Two Rotation (ITR) worm detection and containment algorithm that comprises two sub algorithms. The first sub algorithm called the short term algorithm detects the worm when the failure rate exceeds 100 failures per minute. When the first algorithm fails to detect a worm, the send sub algorithm known as the longer term algorithm is triggered. The long term algorithm detects the worm when the failure rate exceeds 3000 failures per day.

The ITR detection system is deployed at the border routers monitoring the inbound and outbound traffic. The detection of the first failed connection packets triggers the algorithm and the algorithm extracts the source address, source port, destination address, destination port from the packet and creates a record using those information and a counter. The counter keeps track of the number of connection failures between these two nodes. The system maintains two cache tables, one for short term algorithm and the other one for longer term algorithm. The failed count is first stored in the short term cache. When the time interval for the short term algorithm expires and the number of failed connections is below the preset threshold, the value is transferred to the longer term cache for later analysis. The main objective of this arrangement is to distinguish the behavior of worms or infected hosts from normal users. On the critical analysis of the algorithms, it can be see that the short term algorithm works well in identifying vigorous worm but the longer term algorithm would not detect all the stealthy worms as its threshold is set to 3000 failures per day or 2.08 failures per minute on average.

Chen and Tang (2007) have proposed a Distributed Anti-Worm (DAW) architecture that automatically slows down or sometimes totally stops the propagation of a worm within an internet service provider's network. The defense system proposed by these authors identifies the worm infected host from the normal hosts using the behavioral pattern of these hosts. The proposed mechanism relies on the fact that the worm infected host would encounter higher connection failure rate than the normal ones as the infected host scans the Internet randomly, whereas a normal host depends on the Domain Name Systems to resolve a domain name to a valid IP address. The system is composed of two main components, namely the agent that is installed in all the edge routers and a management station that receives data from the agents. The agents monitor the connection failure replies sent to the hosts through the router it is installed in. It identifies the malicious hosts and computes the failure rates. When the failure rates go beyond the preconfigured threshold, the agent starts dropping connections from that host randomly in order to keep its failure rate under control.

The authors have devised two algorithms named, temporal rate-limit algorithm and spatial rate-limit algorithm to manage the worm activity and to bring it down to a low level over a long period while accommodating the temporary aggressive behavior of normal hosts. Agents periodically report the results of the scanning activity and the potential offenders to the management station. A continuous steady increase in the scanning activity is considered as possible worm attack and the worm propagation is curtailed further by dropping more connection requests or stopped altogether by blocking the host that generates a persistently high failure rates. Due to the nature of operation of this mechanism, the strategy may work well on detecting uniform scanning worm and stealthy worms but the non consideration of the impact on the normal network activities due to the operation is major shortcoming. In addition, this mechanism may trigger large number of false alarms and take a long time to detect an actual worm.

Jiang and Zhu (2009) have proposed a new behavioral foot printing approach that complements the signature based approaches. This approach helps users

Table 1: Mechanisms analysis

Author	Worm detection technology (Message error)	Worm detection	Stealthy worm detection	Signature detection	DNA signature detection	DNA filtering	Speed
Schechter *et al.* (2004)	ICMP and RST	✓	-	-	-	-	Slow
Yang *et al.* (2006)	ICMP and RST	✓	✓ but some worm cannot detect it	-	-	-	fast
Chen and Tang (2007)	ICMP and RST	✓	✓	-	-	-	Slow
Jiang and Zhu (2009)	-	-	-	-	-	✓	Fast
Our proposed algorithm	ICMP and RST	✓	✓	✓	✓	Our proposed depends on Jiang filtering	Faster than Xiong algorithm

to detect and profile self propagating worms from the unique worm behavioral perspective. According to the authors, this method uniquely captures a worm's dynamic infection sequences such as probing, exploitation and replication by modeling each interaction step as a behavior phenotype and denoting a complete infection process as a chained sequence. The authors have developed a testbed called vEye and that was used to validate the proposed DNA based algorithm. The vEye is made up of two components, namely a honeyfarm style worm trap network and a virtual worm analysis network. The worm trap network captures live worms and the worm analysis network safely experiments with worms including historical ones.

In this study, we propose the Artificial Immune System (AIS) that can compute the threshold values dynamically. Also, propose an intelligent technique to compute the threshold range for detecting new worms faster. Overall, the proposed algorithm concerns with detection of the worms, stealthy worms, traffic worm signature detection and DNA traffic worm signature detection and send the DNA signature to DNA Filtering system based on the technique proposed by Jiang and Zhu (2009).

Table 1 shows the algorithms proposed by different authors along with their features. From the table it can be seen that the proposed algorithm is better than all the existing methods as it is not only faster than all the other methods but also has employs many techniques complementing each other.

INTELLIGENT FAILURE CONNECTION ALGORITHM (IFCA)

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) (Liu *et al.*, 2012). AIS are one of the most recent approaches in computational intelligence. They provide efficient information processing capabilities (Schaust and Drozda, 2008). IFCA (Rasheed *et al.*, 2010) 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 (Fig. 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. IFCA also ignores the packet when the destination IP is recorded into the counter table because the internet worm attack strategy is "attacking different IP address".

Design IFCA: In IFCA, 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. Assume the failed rate of threshold (β):

$$(\beta) = 100/\text{minute} \quad (1)$$

and source or destination port (X):

$$X = <1...n> \quad (2)$$

and the average of failure connection (AFC) in one minute is:

$$AFC = \text{Counter/Minute} \quad (3)$$

IFCA can calculate the average of failure connection, AFC, after five seconds when IFCA received first failure connection.

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

$$ST = 2^{\wedge} (6.65 + 0.050054 (\beta\text{-}X)) \quad (4)$$

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

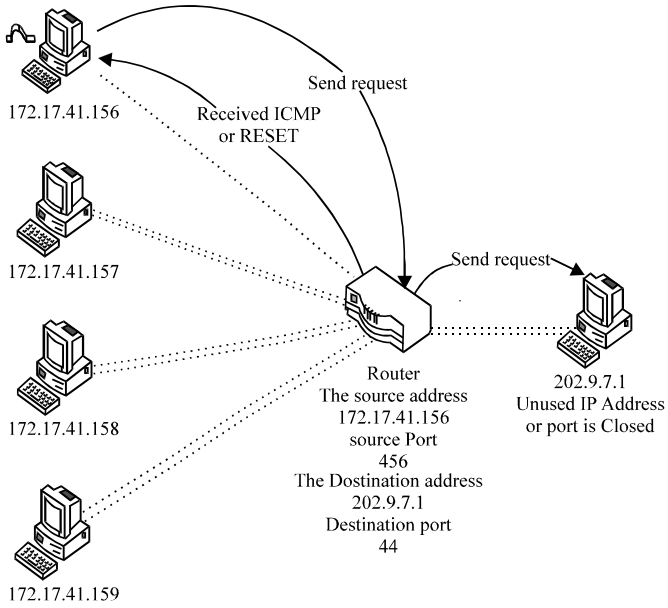

Fig. 3: Error message returned to router

or less 100/min failure connections. Unlike Yang's algorithm, IFCA detects the worm when it is greater than 3000/day failure connections.

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

ST should be greater than fifteen else the traffic will be forwarded. Thus:

$$T1 = ST/AFC \quad (5)$$

$$T2 = \text{Time now-Time start of the algorithm} \quad (6)$$

Unlike Yang's algorithm, IFCA is more dynamic in detecting the worm because it calculates the threshold every time. IFCA detects the worm by comparing T1 to T2 as follows:

```
//worm is detected
if (T2 <= T2) and
(the counter >= ST)
then the worm is detected
else check T1, T2

//give another chance to detect the worm.
if (T2 > T1)
then go to feed back and
decrease the average with new calculation

//a normal counection
if (T1 < T2)
then the traffic will be forwarded
```

During time cumulative computation phase, whenever the counter value does not exceed the threshold the traffic sent from the corresponding IP address would be forwarded as normal activity (Fig. 4).

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

We earlier reports the results for detecting rapid internet worm; then report reports the results for detecting stealthy internet worm. The key parameters used by IFCA are set as follows:

- The total number of vulnerable hosts (N) = 36000
- Number of initial infected hosts I (0) = 1
- Failed rate of threshold (β) = 100/minute
- Rapid Scanning Worm rate = 120/minute
- Stealthy Scanning Worm rate = 2360/day

Detecting rapid scanning worm: In the first experiment of IFCA experiment, Fig. 5 shows the average of failure connections which is 120/minute and the processing time to detect the worm is 25 sec.

Detecting stealthy internet worm: In the second experiment, we used the IFCA to detect the worm. The worm has failure connection of 2360/day and after 30 h IFCA can detect this worm as shown in Fig. 6.

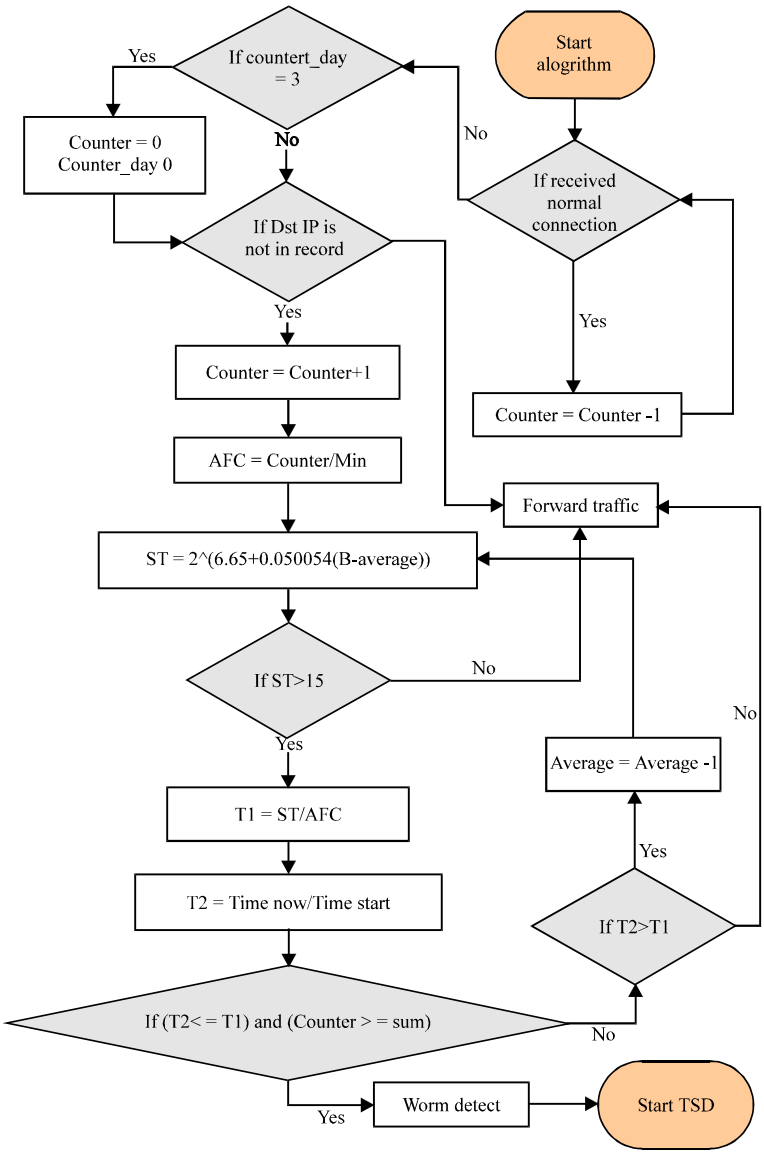

Fig. 4: The flow chart of the IFCA

TRAFFIC SIGNATURE ALGORITHM (TSA)

TSA is a mechanism detects traffic of 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 capturing all the packets synchronization with successful replica from the infector to the victim.

TSA design: TSA process starts when the worm is detected by IFCA, the 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. In this case, our mechanism uses the destination port for packet capturing. Other worms may have different strategy; the source port is fixed like Witty worm but destination port is changed. In this case, our procedure uses the source port for packet capturing. The packet capturing means capturing all packets between infector and victim when the port is opened at the victim side during sending request by the infector computer. The algorithm focuses on 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

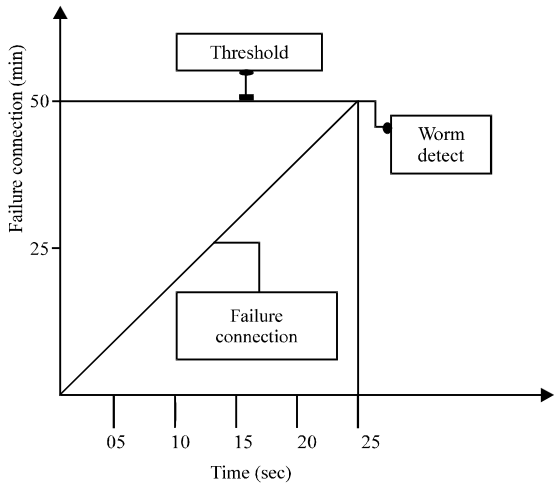

Fig. 5: IFCA detected the worm after 25 sec

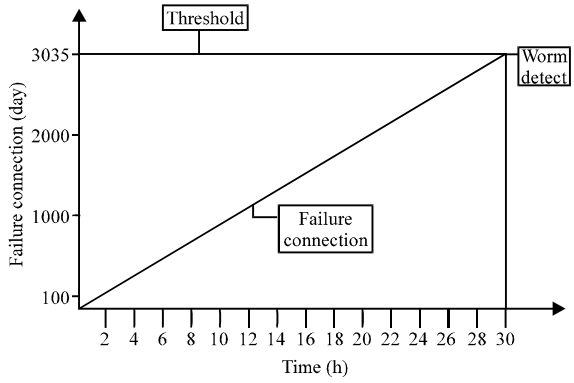

Fig. 6: IFCA detected the worm after 30 h

destination IP is in the record. If the destination IP is in the record that means 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 them.

The algorithm of TSA compares the packets and takes the successful synchronization of the packets that are similar in traffic synchronization(Rasheed *et al*., 2009) as shown in Fig. 7.

Experiments on TSA: Here, we show the experiment synchronization of worm was successful through TSA. The traffic signature of MSBlaster worm is detected by using Ethesnoop program capturing, the traffic signature for MSBlaster 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.

DNA CONVERTER SIGNATURE (DNACS)

First, it will split one infected session into different infection phases, each of which contains a number of traffic (e.g., ICMP, TCP, UDP or connections). Each flow presents a sequence of flow-level actions as elements in the worm's DNA behavior (Jiang and Zhu, 2009). Each DNA character letter in the DNACS describes either a TCP flow with different control bits (SYN (S), ACK (A), RST (R), UDP flow (U), or an ICMP flow (I). The DNA Converter Signature characters synchronization will be sent to all computers that is connected with the router and subsequently to every computer that will filter all input and output packets.

RESULTS

We show the result of MSBlaster worm, where every character of traffic worm is equal to DNA character and is represented as follows:-

S1 = < TCP, X1 /infecter, 135/victim, SYN >
$\overleftarrow{S_1^A}$ = < TCP, 135/victim, X1/infecter, SYN, ACK >
A1 = < TCP, X1/infecter, 135/victim, ACK >
.
.
R1 = < TCP, X1/infecter, 135/victim, RST >
S2 = < TCP, X2/infecter, 4444/victim, SYN >
$\overleftarrow{S_2^A}$ = < TCP, 4444/victim, X2/infecter,SYN, ACK>
A2 = < TCP, X2/infecter, 4444/victim, ACK >
.
.
$\overline{U_1}$ = < UDP, X3/victim, 69/infecter >
U1 = < UDP, 69/infecter, X3/victim >
.
.
R2 = < TCP, X2/infecter, 4444/victim, RST

Then, we use the DNA signature into Jiang's (Jiang and Zhu, 2009) algorithm to filter all input and output packets.

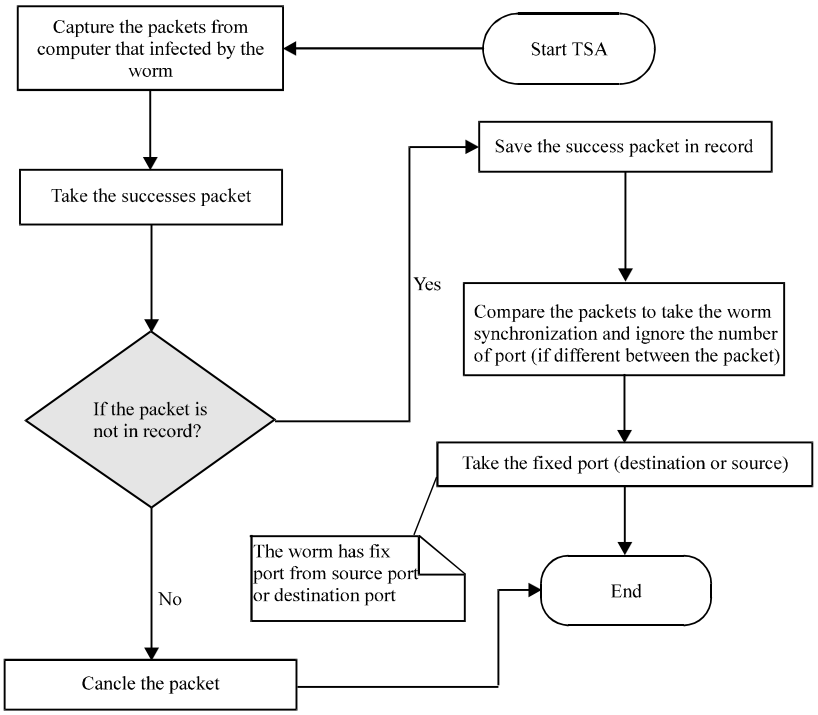

Fig. 7: The flow chart of the TSD

CONCLUSIONS

In this study, we examined only one worm on IFCA and test it on different spreading speeds because all internet worms have same properties of failure connection. DNA Signature can define the worm. Synchronization of the worm is like DNA (Jiang and Zhu, 2009), so we depend on the DNA signature to filter all input and output packets. The worm spread very fast but the techniques to detect the internet worms are slow. Our proposed technique for detecting the worm and generating the signature automatically shows that the algorithm is able to detect the DNA signature for MSBlaster worm.

REFERENCES

Chen, S. and Y. Tang, 2007. DAW: A distributed antiworm system. IEEE Trans. Parallel Distrib. Syst., 18: 893-906.

Costa, M., J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang and P. Barham, 2005. Vigilante: End-to-end containment of internet worms. Proceedings of the 20th ACM Symposium on Operating Systems Principles, October 23-26, 2005, Brighton, UK., pp: 133-147.

Dengyin, Z. and W. Ye, 2010. SIRS: Internet worm propagation model and application. Proceedings of the International Conference on Electrical and Control Engineering, June 25-27, 2010, Wuhan, China, pp: 3029-3032.

Jiang, X. and X. Zhu, 2009. vEye: Behavioral footprinting for self-propagating worm detection and profiling. Knowledge Inform. Syst., 18: 231-262.

Li, P., M. Salour and X. Su, 2008. A survey of internet worm detection and containment. IEEE Commun. Surv. Tutorials, 10: 20-35.

Liu, S., Y. Liu, Y. Tang and R. Jiang, 2012. A novel pattern recognition approach based on immunology. Inform. Technol. J., 11: 134-140.

Min, F. and R. Gupta, 2009. Detecting virus mutations via dynamic matching. Proceedings of the IEEE International Conference on Software Maintenance, September 20-26, 2009, Edmonton, Canada, pp: 105-114.

Mohammed, M.M.Z.E., H.A. Chan, N. Ventura, M. Hashim, I. Amin and E. Bashier, 2010. Detection of zero-day polymorphic worms using principal component analysis. Proceedings of the 6th International Conference on Networking and Services, March 7-13, 2010, Cancun, Mexico, pp: 277-281.

Moskovitch, R., C. Feher and Y. Elovici, 2009. A Chronological Evaluation of Unknown Malcode Detection. In: Intelligence and Security Informatics, Chen, H., C. Yang, M. Chau and S.H. Li (Eds.). Springer-Verlag Berlin Heidelberg, New York, USA., pp: 112-117.

Muda, Z., W. Yassin, M.N. Sulaiman and N.I. Udzir, 2011. A K-means and naive bayes learning approach for better intrusion detection. Inform. Technol. J., 10: 648-655.

Nasir, M.H.N.M., N.H. Hassan and S.S.M. Fauzi, 2008. Protecting windows registry directory and hence increasing the security level of the windows operating system. Inform. Technol. J., 7: 840-849.

Rasheed, M.M., O. Ghazali and N.M. Norwawi, 2010. Server scanning worm detection by using intelligent failure connection algorithm. Res. J. Inform. Technol., 2: 228-234.

Rasheed, M.M., O. Ghazali, N.M. Norwawi and M.M. Kadhum, 2009. A traffic signature-based algorithm for detecting scanning internet worms. Int. J. Commun. Networks Inform. Secur., 1: 24-30.

Rozenberg, B., E. Gudes and Y. Elovici, 2008. A distributed framework for the detection of new worm-related malware. Proceedings of the 1st European Conference on Intelligence and Security Informatics, December 3-5, 2008, Denmark, pp: 179-190.

Schaust, S. and M. Drozda, 2008. Influence of network payload and traffic models on the detection performance of AIS. Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems, June 16-18, 2008, Edinburgh, UK., pp: 44-51.

Schechter, S.E., J. Jung and A.W. Berger, 2004. Fast detection of scanning worm infections. Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection, September 15-17, 2004, Sophia Antipolis, France, pp: 59-81.

Tang, Y. and S. Chen, 2005. Defending against Internet worms: A signature-based approach. IEEE Comput. Commun. Soc., 2: 1384-1394.

Yang, X., J. Lu, Y. Zhu and P. Wang, 2006. Simulation and evaluation of a new algorithm of worm detection and containment. Proceedings of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies, December 4-7, 2006, Taipei, Taiwan, pp: 448-453.

Yoo, S.G., S. Lee, Y. Lee, Y.K. Yang and J. Kim, 2011. Enhanced intrusion detection system for PKMv2 EAP-AKA used in WiBro. Inform. Technol. J., 10: 1882-1895.

Yu, H., M.X. He and H.C. Sun, 2009. The design of firewall in mobile phone based on cross-layer collaboration. Inform. Technol. J., 8: 1049-1053.

Zolkipli, M.F. and A. Jantan, 2010. A framework for malware detection using combination technique and signature generation. Proceedings of the 2nd International Conference on Computer Research and Development, May 7-10, 2010, Kuala Lumpur, Malaysia, pp: 196-199.