A New Generation For Intelligent Anti-Internet Worm Early System Detection

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

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

ABSTRACT

Worm requires host computer with an address on the Internet and any of several vulnerabilities to create a big threat environment. We propose intelligent early system detection mechanism for detecting internet worm. The mechanism is combined of three techniques: Failure Connection Detection (FCD) which concerns with detecting the internet worm and stealthy worm in which computer infected by the worm by using Artificial Immune System; and the Traffic Signature Detection (TSD) which responsible for detecting traffic signature for the worm; and the DNA Filtering Detection (DNAFD) which converts traffic signature to DNA signature and sending it to all computer that connected with the router to create a firewall for new worms. Our proposed algorithm can detect difficult stealthy internet worm in addition to detecting unknown internet worm.

Keywords: *Internet worm Detection, Firewall, Router*.

1.0 INTRODUCTION

The "Morris Worm" of 1988, which required no human mutual action but only a host computer with an address on the Internet and any of several vulnerabilities, created a completely new threat environment (Debany, 2008), that a worm could bring the Internet down in hours. New worm outbreaks have occurred periodically even though their mechanism of spreading was long well understood

Passive worms are different from viruses in that they are completely autonomous entities. Virus is dependent upon a host file or boot sector, and the transfer of files between machines to spread, while a worm can run independently and spread through network connections. Active worm spread in an automated style and can flood the internet in a very short time.

Anti-virus is signature-based technology (Alagna, 2005) which compares the file structure to the signatures stored in its database. If the file contain same signature, so it is infected by the worm. The anti-virus database must be updated continuously to detect new worms.

A computer worm is a self-replicating computer program. It uses a network to send copies of itself to other nodes (computer terminals on the network) and it may do so without any user intervention.

Currently, worms are serious security threat that may cause congestion in the network which leads to large queuing delays, and high packet loss. Since Code Red and Nimda worms were spread in 2001, Epidemic-style attacks have caused huge damages. The Worm handling must be automatic to have any chance of success because worms spread too fast (Costa et al., 2005). The internet is an influential function in the economy and reckon mainstay to the life. Once the internet is broken down, it will cause a huge economic loss.

Unlike viruses, worms do not need to attach themselves to an existing program. Passive worms can run completely independently and through a network of connections, while virus needs a host file, boot sector or file transfer between machines to propagate (worms, 2009).

There are few solutions to solve the worm attack. One of the solutions to update the anti-virus for detects the worms. Anti-virus cannot detect the worm due to its spreading speed. Also, anti-virus cannot detect unknown internet worms automatically because it does not depend on the worm behavior but depends on signature to detect the worm. Routers and firewalls can block packets using traffic signatures, but this happens after the worm has already spread.

Automatic detection is particularly challenging because it is difficult to predict what form that the next worm will take. However, automatic detection and response is fast becoming an imperative because a newly released (flash or topological) worm can infect millions of hosts in a matter of seconds (Staniford, 2004).

The technology is directed to scrutinize the way of the error message, such as RESET in TCP and ICMP (internet controller message protocol) destination unreachable message.

In remainder of this paper is organized as follows. Section 2 describes related work. Section 3 shows design the anti-internet worm through three steps. Section 4 is the conclusion and future work of our system.

2.0 RELATED WORK

Schechter et al. (2004) introduced another worm detection method based on the failed connection. In order to reduce the number of false positive rates, only the first failed connection sent from the forged source IP address to different destination IP address is recorded and normal network activities are also considered. The activities of worms and normal users can be differentiated from the fact that worms usually scan different IP address and produce a larger number of first connection failure packets, and normal users also usually produce first failed connection packets, but will not do like this persistently. This algorithm can detect the internet worm but doesn't work well on detecting stealthy worm. The threshold can't reach to detect stealthy worm.

Yang et al. (2006) built an algorithm for detecting the worm which has two sub algorithms: the first algorithm "short term algorithm" runs well to detect worm while the second algorithm "longer term algorithm" cannot detect all types of the stealthy worm. In addition, Yang's algorithm cannot hold any equations to determine specification when the equation runs in the algorithm to detect early worm if it has higher rate for value in average of failure connection. Yang's algorithm focuses on detecting the computer that contains the worm only.

Jiang & Xu (2006) proposed algorithm to extract a worm's behavioral footprint from the worm's traffic traces. X. Jiang evolution is the number of real worms and their variants confirms the existence of worms' behavioral footprints and demonstrates their effectiveness in worm identification. Jiang algorithm focused to detect known worm by filtering the traffic packet.

Chen & Tang (2007) analyzed the essential character of TCP-based worm's propagation that is sending out a large number of TCP connection requests and proposed an effective approach to detect and contain network worms based on the number of failure connection received by the network routers. The approach can be divided into two defense phrases: short term and longer term. Although this strategy may works well on detecting uniform scanning worm and "stealthy" worm. But the impact of normal network activities has not been considered and then the rate of false alarms could be larger and take time to detect the worm.

Rasheed & Kadhum (2008) proposed two parts sub technique; the first part of the technique is concerned with detecting the internet worm while the second part is concerned with detecting stealthy internet worm. The average of failure connections is the main factor of that method. But this algorithm focused on detecting the computer that is infected by the worm. Our proposed algorithm detects the worm; Stealthy worm detection, Signature detection, and Filter detection (see Table 1).

Table 1: *Mechanisms Analysis*

Algorithm Name	Worm Detection	Stealthy Detection	Signature Detection	Filtering Detection
Schechter (2004)	(X)	-	-	-
Yang (2006)	(X)	(X) but some worm cannot detect it		
Jiang (2006)	-	-	-	(X)
Chen (2007)	(X)	-	-	-
Rasheed (2008)	(X) faster than Yang	(X)	-	-
Our proposed algorithm	(X)	(X)	(X)	(X)

3.0 DESIGN

Our system has three stages:

- Detecting which computer contains the worm.
- Detecting the traffic signature from the computer that infected by the worm.
- Filtering the packet based on firewall's DNA.

Our design steps to create anti-worm as follow:

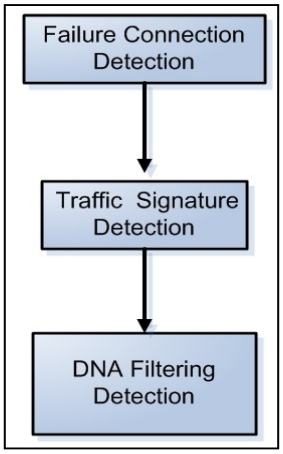

Figure 1: Design Steps

Our mechanism is based on research of behavioral difference between normal user and worm scan. The worm scans different IP addresses per second, normal users usually connect to different IP address and web sites at a slower rate. Especially, normal users maybe have the favorite web sites list, and do not produce so many attempts to connect to random addresses.

The worm generate IP address randomly; for that we received several failure connection when the computer infected by the worm. An ICMP "Destination Unreachable" returned only when the IP addresses is unused (Ellis et al., 2006). See figure 2.

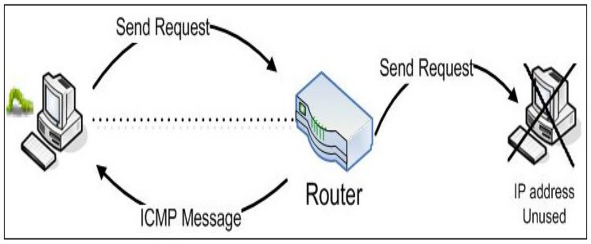

Figure 2: ICMP message

When a SYN packet is sent to a used IP address with destination port closed, TCP RESET packet is returned (Ellis et al., 2006). See figure3.

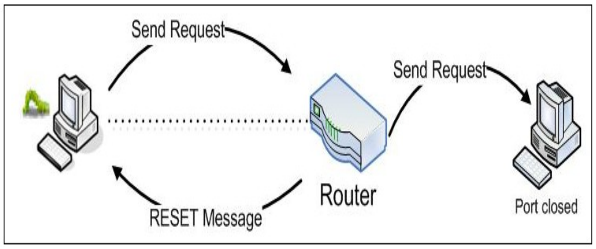

*Figure 3:*RESET message

Our system has a monitor to process for detecting worm signature and sending the signature to all computers that connected by the router (see figure 4).

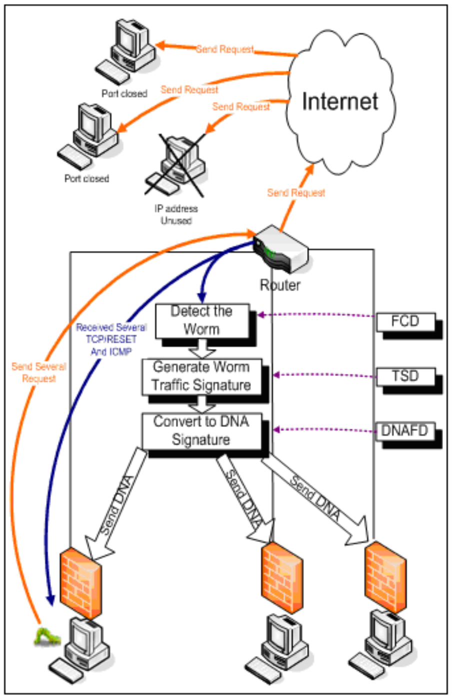

Figure 4: Intelligent Anti-worm system detection

3.1 FAILURE CONNECTION DETECTION (FCD)

FCD appoints the difference between regular connection and worm connection. The worm scans different IP addresses every second. FCD depends on the TCP failure and ICMP unreachable connection on different random addresses. There will be in a large number of failures connections if the computer has worm.

FCD is based on Artificial Immune System; the Artificial Immune System distinguishes 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 effective information processing capabilities. (Schaust & Drozda , 2008).

Our mechanism records the number of first failed connection packets such as ICMP and TCP RESET packets that returned from the external destination address to the internal forged and monitored source IP address based in the router (see figure 5). Once detecting the first failed connection packets, the algorithm then extracts (the source address, source port, destination address, destination port) from the packet and creates the record.

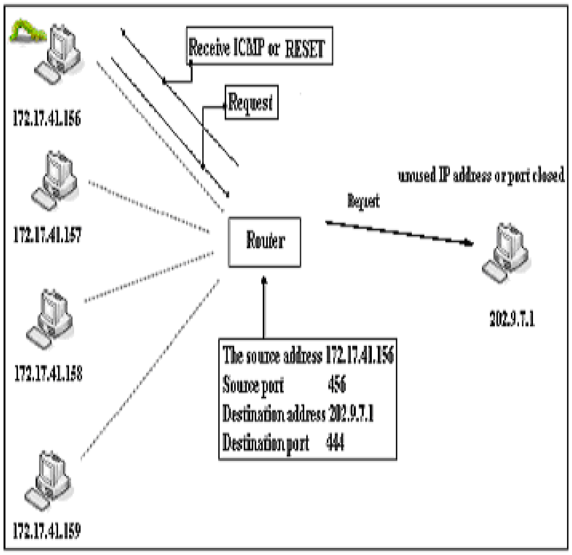

Figure 5: Error message returned to router

The "counter" in the FCD records the first failed connection packets returned from external Destination IP to the internal Source IP during one minute.

We suppose β =100 Then X= (1 to n) average of failure connection in one minute. Threshold can be processed by the following equation:-

$$\textbf{\textit{Summation of threshold}} = 2\text{^} (6.65 + 0.050054 (\beta - X)).$$

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

The Yang's algorithm (Yang et al., 2006) detects the internet worm if the failure connection is equal or greater than 100/minut failure connections by using "long term" algorithm. When the failure connection is equal or greater 3000/day failure connection the Yang algorithm detects this type of stealthy internet worm by using "shorter term" algorithm. Our algorithm can detect the worm by calculating different time on different failure connections. We use Yang's algorithm to calculate the warning.

T1= (summation of Threshold / average of failure connection)
T2= (time now – time start of the algorithm)

Unlike Yang's algorithm, FCD is more dynamic to detect the worm because it calculates the threshold every time.

FCD 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), go to feed back and decrease the average with new calculate to give other chance to detect the worm. If T1 small than T2, then forward the traffic because which means this is normal connection , when the counter value does not exceed the threshold during time cumulative computation phrase, the traffic sent from the corresponding IP address would be forward as normal activity (see figure 6).

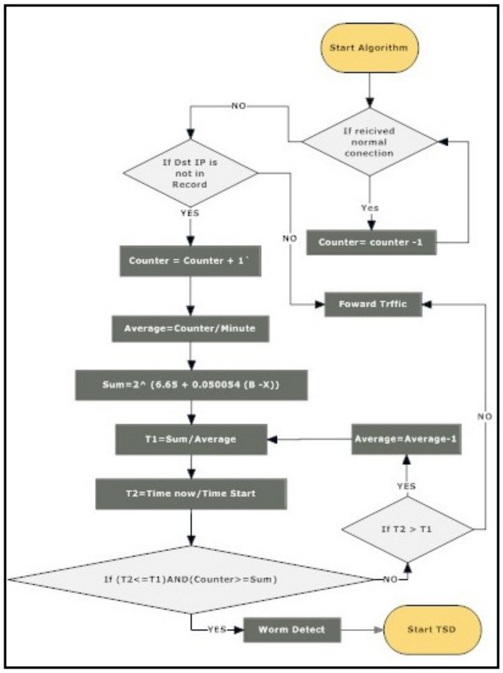

Figure 6: The flow chart of the FCD

3.2 TRAFFIC SIGNATURE DETECTION (TSD)

Our mechanism detects traffic unknown internet worm depending on source IP number that was returned by router so that we can collect the packet by using packets monitor because the worm synchronization is like DNA. So the mechanism depends on the packet filtering using synchronization of internet worm. The Method works when the worm is detected by FCD after that, Signature of internet worm can be detected by using monitor traffic signature.

The algorithm captures all the packets with successful synchronization which started from source port or destination port number. The different infection sequences might have different ports. For example, in the MSBlaster worm the source ports vary with different infection sessions that 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. But other worms may have different strategy; the source port is fixed like Witty worm but destination port is changed. In this case of 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 worm traffic signature (eg. MSBlaster) is represented as follows by using Ethesnoop program capturing:

< 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.

The worm has a successful connection, (see figure 7).
Computer B accepts the worm from computer A because the worm generates random IP address in computer A and the same IP was used for sending to computer B. Under this condition, the worm is transferred from computer A to computer B when the port at computer B was opened.

The method focuses on this type of successful traffic synchronization and captures all these packets by monitor, then send the traffic signature to the DNAFD (see figure 8).

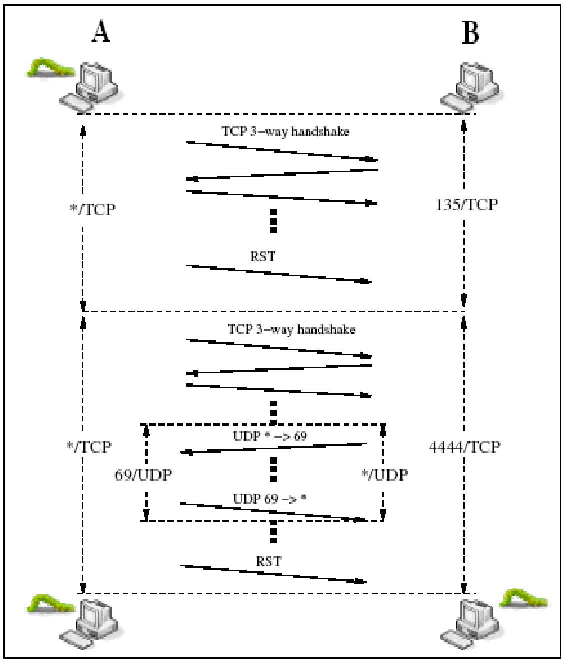

Figure7: Sequence of Infected Worm (Jiang & Xu, 2006)

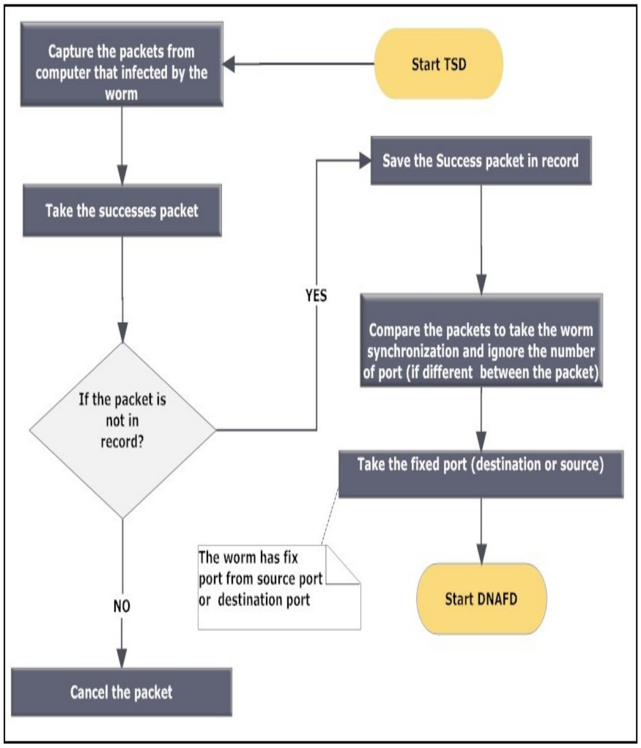
Start TSD
Capture the packets from computer that infected by the worm
Take the successes packet
If the packet is not in record?
YES
Save the Success packet in record
Compare the packets to take the worm synchronization and ignore the number of port (if different between the packet)
Take the fixed port (destination or source)
The worm has fix port from source port or destination port
Start DNAFD
NO
Cancel the packet

Figure 8 : The flow chart of the TSD

3.3 DNA FILTERING DETECTION (DNAFD)

We first break one infection 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 behavioral (Jiang & Xu, 2006). Every sequence of traffic is equal to DNA character and is represented as follows.

S_1 =< TCP, X1 /infecter, 135/victim, SYN >

$\overleftarrow{S_1}^{A}$ =< TCP, 135/victim, X1/infecter, SYN, ACK >

A_1 = < TCP, X1/infecter, 135/victim, ACK >

R_1 =< TCP, X1/infecter, 135/victim, RST >

S_2 = < TCP, X2/infecter, 4444/victim, SYN >

$\overleftarrow{S_2}^{A}$ =< TCP, 4444/victim, X2/infecter, SYN, ACK >

A_2 = < TCP, X2/infecter, 4444/victim, ACK >

$\overleftarrow{U_1}$ =< UDP, X3/victim, 69/infecter >

U_1 =< UDP, 69/infecter, X3/victim >

R_2 = < TCP, X2/infecter, 4444/victim, RST >

Each DNA character letter in the DNA filtering 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 filtering characters synchronization will be sent to all computers that connected with the router and after that every computer will filter all input and output packets (see figure 9).

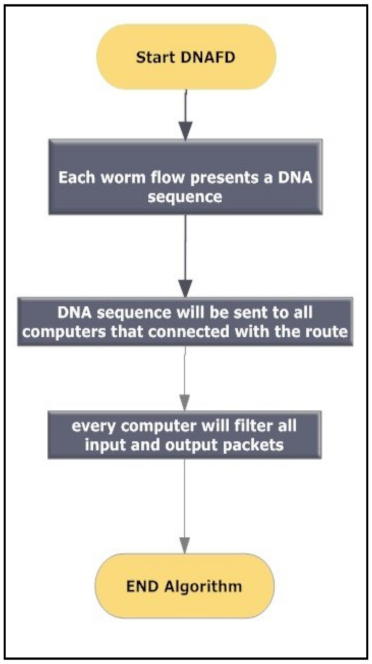
Start DNAFD
Each worm flow presents a DNA sequence
DNA sequence will be sent to all computers that connected with the route
every computer will filter all input and output packets
END Algorithm

Figure 9: The flow chart of the TSD

4.0 CONCLUSION & FUTURE WORK

The worm is very fast spread and the techniques to detect the internet worms are slow. The first stage of our proposed algorithm is detecting the worm that appoints difference between regular connection and worm connection. the second stage is generating the traffic signature for internet worm depends on source IP and source port that was returned by router and third stage represents every sequence of traffic to be equal to DNA character and send DNA characters signature to all computers that are connected by the router, then all computers will filter all input and output packets using DNA. Our future work is to test our system.

REFERENCES

Alagna Tony (2005). Defending the Digital You: How to Fight Online Identity Theft. America, Larstan, Chpter 7.

Chen Shigang & Tang Yong (2007). DAW: A Distributed Antiworm System. *IEEE Journal*, Volume 18, Issue 7, Pages 893 – 906.

Costa M., Crowcroft J., Castro M., Rowstron A., Zhou L., Zhang L., & Barham P. (2005). Vigilante: End-to-end containment of Internet worms. *Proceedings of the 20th ACM Symp. On Operating Systems Principles (SOSP)*, Brighton, UK.

Debany W. (2008). Modeling the Spread of Internet Worms via Persistently Unpatched Hosts. *IEEE Journal*, Volume 22, Issue 2, pages 26 – 32.

Ellis D. , Aiken J., Attwood K., and Tenaglia S.(2004). A behavioral approach to worm detection. Proceedings of the Second ACM Workshop on Rapid Malcode (WORM), pages 49.

Jiang X. & Xu D. (2006). Profiling Self-Propagating Worms via Behavioral Footprinting. *Proceedings of ACM Workshop on Recurring Malcode*, Alexandria, Virginia, USA, Pages 17 – 24.

Rasheed M. & Kadhum M. (2008). Improving the Failure Connection Algorithm for Detecting Unknown Internet Worms. *International Conference on Information Technology and Multimedia 2008 (ICIMU 2008),* Kajang, Selangor DE, Malaysia, Pages 272– 277.

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

Schechte S., Jung J., & Berger A. (2004). Fast Detection of Scanning Worm Infections. *Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID)*, French Riviera, France.

Staniford S. (2004). Containment of Scanning Worms in Enterprise Networks. *Journal of Computer Security*, to appear.

Worms (2009). Computer worms information, Retrieved January 2, 2009, from http://virusall.com/worms.shtml

Yang X., Lu J.,Zhu Y. & Wang P. (2006). 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.