Worm Nonlinear Model Optimization and Feature Detection Technology

Xiaojun Tong and Zhu Wang

School of Computer Science and Technology, Harbin Institute of Technology,
Weihai 264209, China
tong_xiaojun@163.com

Abstract. The static worm propagation model can not accurately describe the propagation of worm. This paper analyzes worm non-linear propagation models, draws out the worm propagation trend and proposes a new dynamic worm non-linear propagation model. Then the worm feature detection technology is designed based on the worm non-linear propagation models. The system uses rule-based detection method to monitor network worms, and gives alarms to server. Experimental results show that the scheme is a good solution to worm detection in multiple network environments and possess with higher detection rate and lower false alarm rate.

Keywords: IDS, Worm, Worm model optimization, Feature detection.

1 Introduction

As the computer and internet technology are continuously developing, the open resources and sharing of information have brought us great conveniences but also brought us the security problems. The network worm attacks are on the top of the list among varieties of network security threats.

The models research of worm propagation is hot fields. The ideal propagation model reflects the worms' propagation activities effectively and identifies the weaknesses of the worm propagation circle. At the same time, it can also forecast the potential threats brought by the worms and provides instructions for worm detecting.

The routing-worm propagation model in the IPv6 network has been proposed in Reference [1]. Based on the IPv6 network environment, it analyzed the scanning strategy of routing worm-IPv6 and simulated the propagation trends of Routing Worm-IPv6 via Two-Factor model. The model of anti-worms against malignant-worms[2] indicates that if the anti-worms adopts some control strategies, it can achieve a satisfactory effect in resisting malignant-worms, such as specifying the activity time, specifying the spread range, specifying the amount of copies and the slow-spreading mechanism.

There are many detection models in response to the large-scale and swift worm propagation[3-8]. A worm detection algorithm CWDMLN was proposed in Reference [3] , which makes use of the local network's cooperation and analyzes some worm's propagation features. The algorithm supplies alarms for worms' intrusion according to

P. Sénac, M. Ott, and A. Seneviratne (Eds.): ICWCA 2011, LNICST 72, pp. 156–169, 2012.

the worms' petal-like communication mode and invalid connections by deploying honey-pot in the LAN. Although it is feasible in the LAN, it is helpless in extensive detecting in multiple networks. In reference [3], the author comes up with some suggestions for the improvement but hasn't realized it yet. Reference [4] has brought up a distributed worm containment mechanism. Although the computational overhead is small and detection rate is high, but such detection mechanisms must be deployed on the router and does not apply to small and medium networks in general as the environmental requirements are too high.

In this paper, we analysed detail worm non-linear propagation model and proposed a new optimization model according to the adding of parameters, designed a distributed fusion-worm-detection system which has great practical significance on detecting large scale worm propagation and on limiting the damage to the network.

The paper is constructed as follows. First, we analyzed the worm features and work method of worms. Then, we did some researches on several classical non-linear worm propagation model, proposed the non-linear propagation optimization of worm and designed the distributed worm detection model . We did some experiments to verify it. Finally, we conclude the paper.

2 Analysis of Worm Non-linear Model

2.1 Feature of Worm

Network worm is usually a standalone program which runs without any user intervention. It spreads itself to other computers in the same LAN which has vulnerabilities. While the virus is a program or programming code that can graft its copying onto another program including the operating system. The virus can not run automatically, it needs to be activated by the host program [9]. Both the computer worm and virus can replicate and can spread themselves, which makes it's difficult to distinguish them. Especially, in recent years, more and more virus come to use worms' technology[10-11]. Meanwhile, worm adopts the virus technology too. So it is of great necessity to distinguish and to analyze their features (showed in Table 1).

Table 1. Differences between virus and worm

Item	virus	worm
state of existence	parasitism	independent entity
replication form	insert into a file	replicate itself
transmission mechanism	activated by host program	system vulnerability
targets	local files	other hosts on the network
trigger	computer users	program itself
mainly influence	files , system	network and system performance
precautionary measures	remove from the host file	patch for the system, firewall

2.2 Work Flow of Worm

Worm is a kind of intelligent and automatic program [12-14]. Its working process is divided into four stages seen in Fig.1: scanning, penetration attack, on-site processing and replication. First, the infected hosts attempt to pick up the victims with vulnerabilities hosts for infection. Secondly, the worm sends packets to the victims to carry out the penetration attack. Thirdly, the worm does the on-site processing then hides itself and collects information, the aim is to make sure that the victims have no ware of being infected so that it is to cause more serious damage. Finally, during the self-replication stage, worm produce copies itself and repeats the steps above. The work flow of worm is seen in Fig.1.

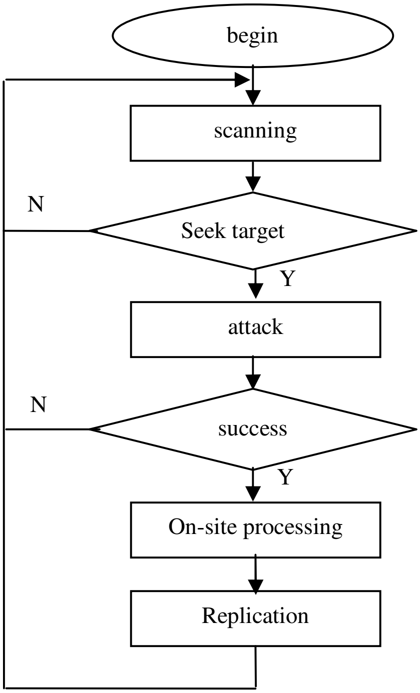


Fig. 1. Flow of worm work

2.3 Analysis of Worm Non-linear Propagation Model

Our research and analyses are based on several classical propagation models, which respectively are Simple Epidemic Model, Kermack-Mckendrick model, and the Two-Factor model.

SEM Model. In the simple epidemic model we divide the hosts into two groups: susceptible hosts and infective hosts. The model assumes that once a host is infected by a worm, it will stay in its infectious state forever, which means that the state of a host must be either susceptible or infective. The simple epidemic model for a finite population is as follows:

$$\frac{dJ(t)}{dt} = \beta J(t)[N - J(t)] \tag{1}$$

where " $J(t)$ " is the number of the infective hosts at time t . "N " is the size of hosts. β is the infection rate. At the beginning,

Kermack-Mckendrick Model. Different from the simple epidemic model, Kermack-Mckendrick model considers the removal process of infectious hosts. So there are totally three states of the vulnerable hosts: susceptible, infective and immune. Hosts in "immune" state can not infect other hosts forever.

Set $I(t)$ is the number of the infective hosts at time t . $R(t)$ is the number of immune hosts at time t . While $J(t)$ is the number of hosts which have been infected by time t .

So that we get the equation:

$$J(t) = I(t) + R(t) \tag{2}$$

The Kermack-Mckendrick model：

$$\begin{cases} dJ(t)/dt = \beta I(t)[N - J(t)] \\ dR(t)/dt = \gamma I(t) \\ J(t) = I(t) + R(t) = N - S(t) \end{cases} \tag{3}$$

Where β is the infection rate. γ is the immune hosts' removed rate from the hosts infected. $S(t)$ is the number of susceptible hosts at time t . N is the total number of the vulnerable hosts.

The Two-Factor Model. There are several dynamic parameters to be assured: $\beta(t)$ 、 $R(t)$ and $Q(t)$. $\beta(t)$ is the infection rate which changes with time. $R(t)$ is the number of removed hosts from infective ones at time t . $Q(t)$ matches the number of removed hosts from susceptible ones at time t . So between the time t and $t + \Delta t$, the change of the number of susceptible hosts is:

$$S(t + \Delta t) - S(t) = -\beta(t)S(t)I(t)\Delta t - \frac{dQ(t)}{dt}\Delta t \tag{4}$$

where $s(t)$ is the number of susceptible hosts at time t . $I(t)$ is the number of infective hosts at time t .

The susceptible hosts' immunity process is described as follow in the Two-Factor model:

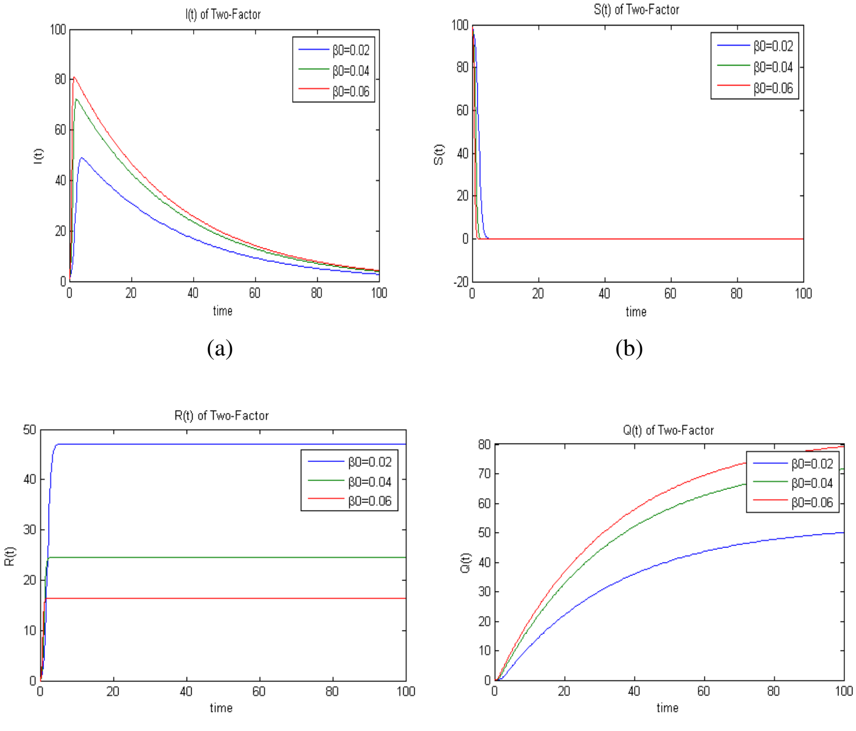
$$\frac{dQ(t)}{dt} = \mu S(t)J(t) \tag{5}$$

Based on the Two-Factor-Model's assume of the dynamic properties, the complete differential equations are as follows:

$$\begin{cases} dS(t)/dt = -\beta(t)S(t)I(t) - dQ(t)/dt \\ dR(t)/dt = \gamma I(t) \\ dQ(t)/dt = \mu S(t)J(t) \\ \beta(t) = \beta_0[1 - I(t)/N]^\eta \\ N = S(t) + R(t) + I(t) + Q(t) \\ I(0) = I_0 << N; S(0) = N - I_0; R(0) = Q(0) = 0; \end{cases} \tag{6}$$

where γ is the infective hosts' immunity, $J(t) = I(t) + R(t)$ describes the number of hosts which have been infected now or before. μ is a constant, $\mu J(t)$ is the immunity rate of susceptible hosts at time t.

β_0 is the initial value of infection rate. The exponent η is used to adjust the infection rate sensitivity to the number of infective hosts $I(t)$.



(a) (b)

(c) (d)

Fig. 2. $I(t), S(t), R(t), Q(t)$ of Two-Factor Model

If we set $\mu = 0$, $\eta = 0$ and $\gamma = 0$, we can get the SEM from the Two-Factor-Model. If we set $\mu = 0$ 、 $\eta = 0$, and $\gamma \neq 0$, we get KM from the Two-Factor-model.

Set $\gamma = 0.03$, $N = 100$, $I(0) = 1$, $R(0) = 0$, $Q(0) = 0$, $\mu = 0.01$, $\sigma = 3$, and β respectively are 0.02, 0.04, 0.06. According to the formula (9) and (11), we can get the function of $I(t)$, $S(t)$, $R(t)$, $Q(t)$ and t as follows in Fig. 2

The Two-Factor-Model is the extension of the SEM and KM. It makes up for the shortage of the two models and is more suitable to describe the network worm's propagation model. But the Two-Factor-Model doesn't take the large-scale automatic patches or upgrades into consideration. Similarly, it also considers γ as a constant. Since that the Two-Factor-Model believes that the immunity rate of the susceptible hosts is increasing with time, the immunity rate of the infective hosts also should be increasing with time. So it is not appropriate to take γ as a constant.

2.4 Improved Two-Factor Model

The two-factor model already can describe the worms' propagation well. But there still are some flaws. Although the two-factor has brought the susceptible and infected hosts' immune into consideration, but the immune rate is constant here which is not in accordance with the actual network situation.

We can consider the problem from two aspects:

First, the infected hosts' immunity γ. The infected hosts can eliminate the worms by downloading patches, anti-virus software or network administrator's intervention. And at the same time, the infected host can get his immunity. Here γ is exactly the parameter describing the ability of infected hosts' immunity. Parameter γ is infected by many factors such as the total number of the immunity hosts, the official patches and corresponding anti-virus software and so on. So γ must be a variable which changes with time. In the early period, γ is quite low and later it begins to grow when there are more and more immunity hosts in the network or the official patches released. For that reason, we need to find a propitiate model to describe γ dynamic features in order to explore the worms propagation in actual situation more accurately.

Secondly, the susceptible hosts' immunity μ . Taking the susceptible hosts' immunity into consideration is a great improvement of two-factor model compared with the SEM. But two-factor only considers the perfect condition which μ is a constant through all the worm propagation process, which is obviously not according to the actual situation. In order to understand the dynamic features of μ , let us propose a question first. How do the susceptible hosts get their immunity? We can explain the question from two aspects:

A, the susceptible host gets information about the worm or official patches or even anti-virus software from other immunity hosts in the same network. So it gets its immunity.

B, the susceptible host itself gets in touch with the network monitor center and gets the corresponding anti-virus software.

For the last two reasons, μ must be a variable and its evolution must be similar to γ's. In the early time, μ is quite low as there are few immunity hosts in the network.

While with the time going, more and more immunity hosts emerge and μ begins to grow, meanwhile the ability of susceptible hosts' immunity grows too.

After a mount times of experiment, we conclude the math model of μ and γ as Fig.3 and Fig.4.

$$\gamma = \gamma_0 (1 + R(t) / N)^{\sigma} \tag{7}$$

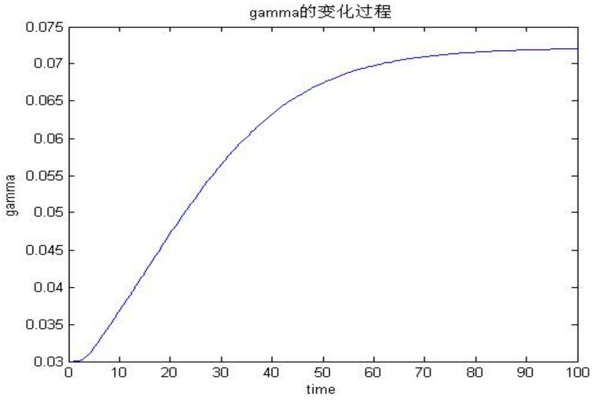


Fig. 3. The model of γ

$$\mu = \mu_0 (1 + Q(t) / N)^{\sigma} \tag{8}$$

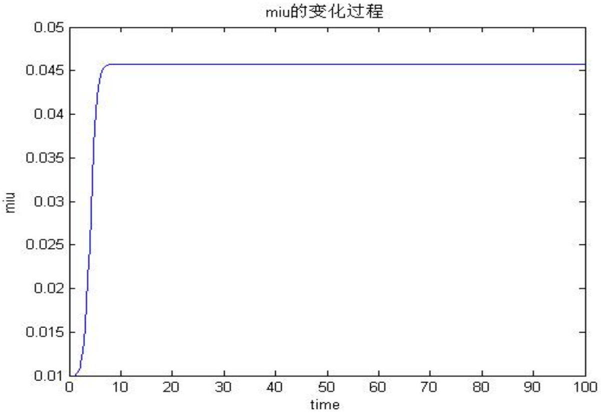


Fig. 4. The model of μ

The worm propagation trend of the modified model is same to primary Two-Factor model. As shown in the fig.5 and in Fig.6.

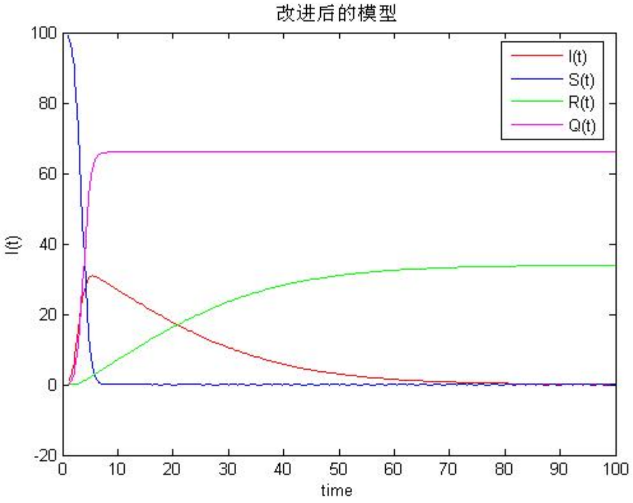


Fig. 5. Improved model of Two-Factor

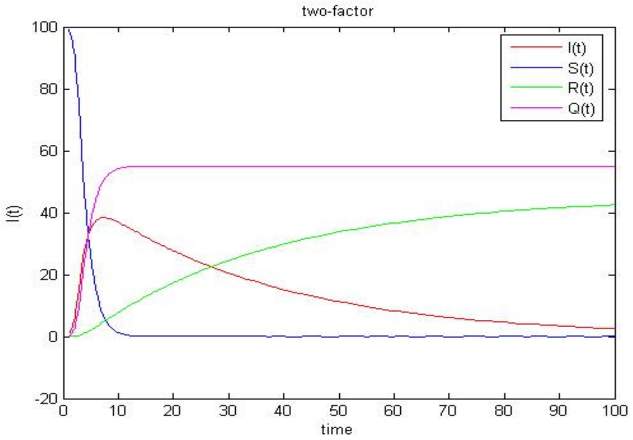


Fig. 6. The model of Two-Factor

After simulation by MATLAB, the improved two-factor model's overall trend is the same with the original model.

The difference is that the modified model can return to normal more quickly than the primary model, which is conform to the actual situation perfectly. Because after a certain worm breaks out, the network administrator and the clients will take all kinds of measures to restrain the worm's propagation as soon as possible, so that the network can recover to the normal situation.

We can conclude from the trend figure that there are two improvements of the improved model:

1. A certain worm breaks out in a network with 100 hosts. There are totally 38 hosts get infected in the original model while 30 hosts exist in the improved model. Compared with the original model, the total infected host is 8 percentages lower than the original ones, which is according with the actual situation well.

2. The improved model reaches its peak 6 seconds after the worm breaks out and then the infected hosts decrease sharply. The original model's peak is about 10 seconds after the worm break out and infected hosts decrease slowly. For that reason, the improved model can describe the worm propagation more accurately. With the improvement of the technology, quickly information communication and people's awareness of worms, the time taken to defeat worm must be shorter and shorter.

3 Worm Feature Detection Model of Worms

3.1 Common Intrusion Detection Framework

In recent years, the intrusion detection has been greatly developed. The Defense Advanced Research Projects Agency (DARPA) together with the Intrusion Detection Working Group of Internet Engineering Task Force (IERF) have set the standard criterion of IDS and bring up the Common Intrusion Detection Framework (CIDF) showed in Fig.7 as follows:

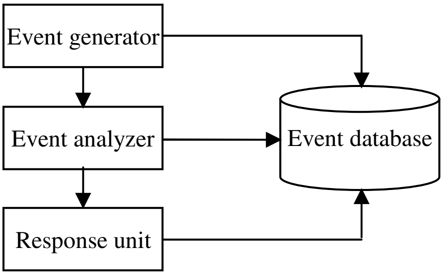


Fig. 7. Architecture of CIDF

The event generator picks information which interests it and transforms the information into the standard form so that other components in the system can use. The event analyzer analyses events and do core-intrusion-detection and then creates new GIDOS. The response unit decides the measures which should be taken according to the new GIDOS.

3.2 Design of Worm Rules

Through some researches on several common worms, we get the rules of them which have been detected in our detection system as follows.

（1）Rules of Ramen Worm

Alert tcp $HOME_NET any -> $EXTERNAL_NET 27374 (msg::”MISC ramen worm”;flow:to_server , established; content:”GET ”; depth:8; nocase;reference:arachnids , 461; classtype:bad-unknown; sid:514;rev:5;)

（2）Rules of CodeRed Worm

Alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:”WEB-IIS CodeRed v2 root.exe access”; flow:to_server , established; uricontent:”/root.exe”; nocase; reference: url , www.cert.org/advisories/CA-2001-19.html; classtype:web-application-attack; sid:1256;rev:8;)

（3）Rules of Slapper Worm

Alert udp $EXTERNAL_NET 2002 -> $HTTP_SERVERS 2002 (msg:”MISC slapper worm admin traffic”; content:”|00 00|E|00 00|E|00 00|@|00”; depth:10; reference:url , isc.incidents.org/analysis.html?id=167; reference:url, www.cert.org/advisories/CA-2002-27.html; classtype:Trojan-activity; sid:1889; rev:5;)

（4）Rules of Slammer Worm

Alert udp $HOME_NET any -> $EXTERNAL_NET 1434 (msg:”MS-SQL Worm propagation attempt OUTBOUND”; content:”|04|”; depth:1; content:”|81 F1 03 01 04 9B 81 F1|”; content:”sock”; content:”send”; reference: bugtraq, 5310; reference:bugtraq, 5311; reference:cve, 2002-0649; reference:nessus , 11214; reference: url , vil.nai.com/vil/content/v_99992.htm; classtype:misc-attack; sid:2004;rev:7;)

（5）Rules of ACworm Worm

Alert tcp $EXTERNAL_NET any -> HTTP_SERVERS $HTTP_PORTS (msg:”WEB-MISC Apache Chunked-Encoding worm attempt”; flow:to_server, established; content:”CCCCCCC|3A|AAAAAAAAAAAAAAAAAAA”; nocase; reference:bugtraq, 4474; reference:bugtraq, 4485; reference:bugtraq, 5033; reference:cve, 2002-0071; reference:cve, 2002-0079; reference:cve, 2002-0392; classtype:web-application-attack; sid:1809; rev:9;)

（6） Rules of Sadmind Worm

Alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:”WEB-MISC sadmind worm access”; flow: to_server, established; content:”GET x HTTP/1.0”; depth:15; reference:url, www.cert.org/advisories/CA-2001-11.html; classtype:attempted-recon; sid:1375; rev:6;)

（7） Rules of Code Red II Worm

Alert tcp $external_net any -> $http_net $http_ports (msg:”Web-IIS ISAPI.ida attempt”; uricontent:”.ida?”; nocase; dsize: 239; flags:A+;)

（8） Rules of Nachi Worm

Alert icmp $HOME_NET any -> $ EXTERNAL_NET any (msg:”Nachi”; content:”|aaaaaa|”; dsize:64; itype:8; offset:1; depth:6; reference:arachnids, 154; sid:483; classtype:misc-activity; rev:2;)

（9） Rules of Witty Worm

alert udp any 4000 -> any any (msg:"ISS PAM/Witty Worm Shellcode"; content:"|65 74 51 68 73 6f 63 6b 54 53|"; depth:246; classtype:misc-attack; reference:url,www.secureworks.com/research/threats/witty; sid:1000078; rev:1;)

（10） Rules of Lion

① BIND infoleak root uses follows rules ：

Alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:”IDS482/named-exploit-infoleak-lsd”;content:”|AB CD 09 80 00 00 00 01 00 00 00 00 00 00 01 00 01 20 20 20 20 02 61|”; sid:1000081;rev:1;)

② BIND8 TSIG buffer overflow uses follows arachNIDS rules ：

Alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:”IDS489/named-exploit-tsig-lsd”;content:”|3F 90 90 90 EB 3B 31 DB 5F 83 EF 7C 8D 77 10 89 77 04 8D 4F 20|”; sid:1000082;rev:1;)

4 The Analysis of Experiment Results

Considering the security of system, the paper uses self-developed simulation program to send worms that meet specific characteristics of the worm packets. Here mainly to simulate the three kinds of worms: Witty worm, Slammer worm and Ramen worms. They represent the three kinds of worms of UDP and TCP protocols using two types of worm spread .

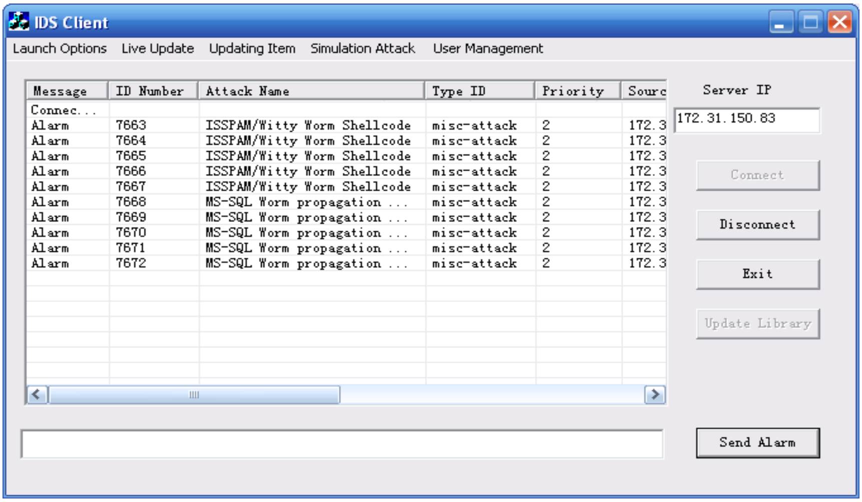
Slammer worm itself is packaged in a size of 376 bytes of UDP packets from any source port and is sent to any address on the network host port 1434 of UDP.

If the SQL Server Resolution Service of host opens and does not install the appropriate patch program, the worm will use the buffer overflow vulnerability to infect them. The first byte of Slammer worm packet is 0x04, in which has the contents of 0x810xF10x030x010x040x9B0x810xF1, ‘sock’ and ‘send’ content.

The processes of experiment are as follows:

(1) Start the console and set the state for waiting connection of the worm detection end.

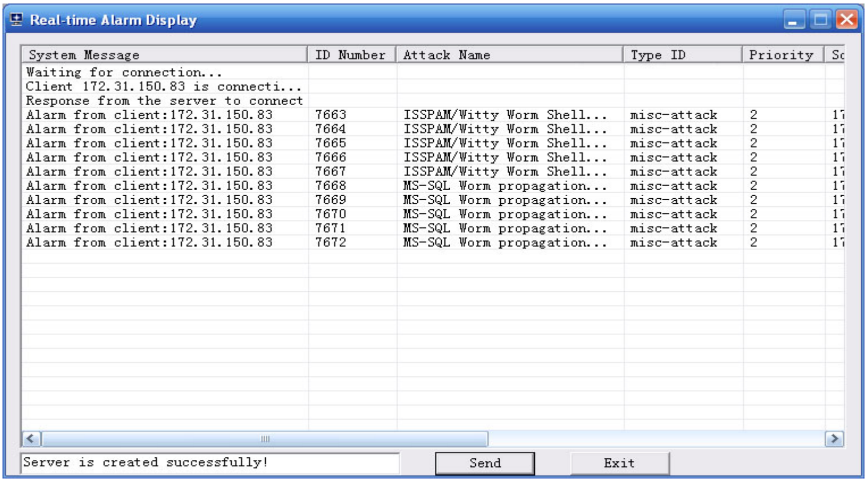
(2) Start worm detection end and connect it to the console.



Message	ID Number	Attack Name	Type ID	Priority	Sourc
Connec...					
Alarm	7663	ISSPAM/Witty Worm Shellcode	misc-attack	2	172.3
Alarm	7664	ISSPAM/Witty Worm Shellcode	misc-attack	2	172.3
Alarm	7665	ISSPAM/Witty Worm Shellcode	misc-attack	2	172.3
Alarm	7666	ISSPAM/Witty Worm Shellcode	misc-attack	2	172.3
Alarm	7667	ISSPAM/Witty Worm Shellcode	misc-attack	2	172.3
Alarm	7668	MS-SQL Worm propagation ...	misc-attack	2	172.3
Alarm	7669	MS-SQL Worm propagation ...	misc-attack	2	172.3
Alarm	7670	MS-SQL Worm propagation ...	misc-attack	2	172.3
Alarm	7671	MS-SQL Worm propagation ...	misc-attack	2	172.3
Alarm	7672	MS-SQL Worm propagation ...	misc-attack	2	172.3

Server IP: 172.31.150.83

Fig. 8. Witty worm detected on client end



System Message	ID Number	Attack Name	Type ID	Priority	Sc
Waiting for connection...					
Client 172.31.150.83 is connecti...					
Response from the server to connect					
Alarm from client:172.31.150.83	7663	ISSPAM/Witty Worm Shell...	misc-attack	2	17
Alarm from client:172.31.150.83	7664	ISSPAM/Witty Worm Shell...	misc-attack	2	17
Alarm from client:172.31.150.83	7665	ISSPAM/Witty Worm Shell...	misc-attack	2	17
Alarm from client:172.31.150.83	7666	ISSPAM/Witty Worm Shell...	misc-attack	2	17
Alarm from client:172.31.150.83	7667	ISSPAM/Witty Worm Shell...	misc-attack	2	17
Alarm from client:172.31.150.83	7668	MS-SQL Worm propagation...	misc-attack	2	17
Alarm from client:172.31.150.83	7669	MS-SQL Worm propagation...	misc-attack	2	17
Alarm from client:172.31.150.83	7670	MS-SQL Worm propagation...	misc-attack	2	17
Alarm from client:172.31.150.83	7671	MS-SQL Worm propagation...	misc-attack	2	17
Alarm from client:172.31.150.83	7672	MS-SQL Worm propagation...	misc-attack	2	17

Server is created successfully!

Fig. 9. Worm Alarm received on console end

(3) Start the host A and host B and run normal TCP and UCP applications on them.

(4) Run the worm simulation program on host A and host B. The program generates worm packets such as Slammer worm, Witty worm or Ramen worms. The detection side detected the corresponding worm data packets and generated alerts as shown in Fig. 8. At the same time the console receives the worm alert information, as shown in Fig. 9.

5 Conclusions

This paper analyzed the model of worm non-linear propagation, and proposed a nonlinear model optimization of worm and designed a distributed detection system of worm. Experiments have proven that the new model can accurately reflect the propagation trend of the worm, and the worm feature detection system not only is able to achieve a high detection rate, but also be able to carry out a wide range of network monitoring. The system has high detection rate and low false alarm rate can be applied to worm detection..

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant No. 60973162), the Natural Science Foundation of Shandong Province (Grant No. ZR2009GM037), the Science and technology of Shandong Province of China (Grant No.2010GGX10132), the Scientific Research Foundation of Harbin Institute of Technology at Weihai (Grant No. HIT(WH) ZB200909).

References

1. Xu, Y.-g., Qian, H.-y., Li, H.-f.: Routing worm propagation model in IPv6 networks. Application Research of Computers 3920 (2009)
2. Zhang, D.-x., Peng, J., Hong, H.E.: Research the Propagation Model of Internet Worm. Network Communication and Security, 1244–1246 (2007)
3. Zhang., X.-y., Qing, S.-h., Li, Q., Li, D.-z., He, C.-h.: A Coordinated Worm Detection Method Based on Local Nets. Journal of Software, 412–421 (2007)
4. Zhao, g., Zhang, t.: Design of the worm detection system Based on the worm propagation characteristics. Computer Security, 114–118 (2009)
5. Ram, D., Cangussu, W., Sudeep, P.: Fast Worm Containment Using Feedback Control. Dependable and Secure Computing 5(2), 119–136 (2007)
6. Kim, H.A., Karp, B.: Autograph. Toward Automated, Distributed Worm Signature Detection. In: Proceedings of the 13th USENIX Security Symposium, San Diego, CA, pp. 59–66 (2004)
7. Dagon, D., Qin, X., Gu, G., Lee, W., Grizzard, J.B., Levine, J.G., Owen, H.L.: HoneyStat: Local worm detection using honeypots. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 39–58. Springer, Heidelberg (2004)
8. Tang, Y., Chen, S.: Defending against internet worms: A signature-based approach. In: Proceedings of IEEE INFOCOM 2005, Hong Kong, pp. 13–23 (2005)
9. Eugene, H.: The Internet worm programs. ACM Computer 23(3), 17–57 (1989)

10. Wang, Y., Wang, C.X.: Modeling The Effects of Timing Parameters on Virus Propagation. In: Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM 2003), pp. 61–66. ACM press, Washington (2003)
11. Dantu, R., Cangussu, J., Yelimeli, A.: Dynamic control of worm propagation. Information Technology 1(3), 419–423 (2004)
12. Streftaris, G., Gibson, G.J.: Statistical Inference for Stochastic Epidemic Models. In: Proceedings of the 17th International Workshop on Statistical Modeling, China, pp. 609–616 (2002)
13. Zou, C.C., Gong, W., Towsley, D.: Code Red Worm Propagation Modeling and Analysis. In: Proceedings of the 9th ACM Symp on Computer and Communication Security, pp. 138–147. ACM Press, Washington (2002)
14. Bishop, M.: A Model of Security Monitoring. In: Proceedings of Fifth AnnualComputer Security Applications Conference, New Orleans, pp. 249–251. IEEE Computer Society, Washington DC, USA (1989)