Computers & Security (2005) **24**, 334–346

ELSEVIER

Computers & Security

www.elsevier.com/locate/cose

A survey and trends on Internet worms☆

Sihan Qing[a,b,c], **Weiping Wen**[a,b,c,*]

[a]*Institute of Software, Chinese Academy of Sciences, Beijing 100080, China*
[b]*Engineering Research Center for Information Security Technology, Chinese Academy of Sciences, Beijing 100080, China*
[c]*Graduate School of Chinese Academy of Sciences, Beijing 100080, China*

Received 23 May 2004; revised 2 October 2004; accepted 5 October 2004

KEYWORDS
Network security;
Internet worms;
Scanning strategies;
Propagation model

Abstract With the explosive growth and increasing complexity of network applications, the threats of Internet worms against network security are more and more serious. This paper presents the concepts and research situations of Internet worms, their function component, and their execution mechanism. It also addresses the scanning strategies, propagation models, and the critical techniques of Internet worm prevention. Finally, the remaining problems and emerging trends in this area are also outlined.

Introduction

With the explosive growth of Internet applications, the threats of Internet worms against computer systems and network security are increasingly serious. Especially in the environment of the Internet, various ways of the worm propagation and the complexity of the application environment result in much higher frequency of outbreak, much deeper latency and more wider coverage of Internet worms. ''Morris'', a well-known worm appearing in 1988, was the first Internet worm incident known to us (Spafford, 1988). Since then, Internet worms have been a main issue faced by computer security researchers. Internet worms are gaining more attention again because of the outbreak of the worm ''CodeRed'' in July, 2001 (EEye Digital Security, Code Red Worm; CERT, 2001).

Currently the research on Internet worms mainly focuses on the function structure, execution mechanism, scanning strategies, propagation model, countermeasure technology, etc. Spafford (1988) was the first person to analyze the structure and the function mechanism of the worm ''Morris''. Weaver from University of California, Berkeley, investigated the quick scanning strategies of worms and implemented the worm

☆ Supported by the National Natural Science Foundation of China under Grant No. 60083007; the National Grand Fundamental Research 973 Program of China under Grant No. G1999035810.

* Corresponding author. Institute of Software, Chinese Academy of Sciences, Engineering Research Center for Information Security Technology, Beijing 100080, China.
E-mail addresses: qsihan@ercist.iscas.ac.cn (S. Qing), qing1010@ercist.iscas.ac.cn (W. Wen).

0167-4048/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2004.10.001

''Warhol'' (Weaver, 2002; Staniford et al., 2002; Weaver, Warhol worms) in experiments. He also theoretically concluded that the worm had the ability to infect throughout the Internet in 30 min. He also emphasized, to resist the worm attack, the importance of the automation of detection, analysis, and response. In terms of the propagation model, Kephart et al. of IBM investigated the virus propagation model from 1991 to 1993 (Kephart et al., 1993; Kephart and White, 1993). Based on their work, Zou et al. (2002) analyzed a differential equation based Two-Factor worm propagation model for the worm ''CodeRed''. In terms of anti-worm technology, White in IBM thought that the traditional anti-virus techniques on a single computer were no longer applicable to the prevention of worms (Steve, 1998). In 2000, IBM initiated an anti-worm project, and attempted to develop an environment of software and hardware to automatically detect and prevent the worms (Arnold et al., 2000). Song et al. (2001) worked on the statistical properties of network throughput resulting from Internet worms and attempted to prevent Internet worms through the detection of abnormal Internet traffic. Moore et al. (2003a) proposed three factors to evaluate the validity of anti-worm prevention system: response time, containment strategy, and deployment scenario. He thought that these three parameters were difficult to be satisfied in most current anti-worm systems.

In recent years, governments and research organizations have all recognized the importance of the study of Internet worms. The US government invests about 546 million dollars in building up a network attack test bed to investigate worm and virus in University of California, Berkeley. The test bed is composed of more than 1000 computers (Yang and Relations, NSF awards). Staniford et al. (The worm information center) set up a technical website on worm research and publicized the research results periodically. ''WORM 2003'' conference was held in Washington, DC in October, 2003. The conference discussed the past, the present, and the future of Internet worms, the classification of computer worms, the simulation of worm traffic, the design and test of a worm warning system, the simulation of propagation strategy, the technology of anatomy and separation of worm model, etc. In China, the researches on Internet worms gain more and more attention. Governments and security companies are actively engaged in preventing and cleaning the worm. In the field of the research of Internet worms, according to literature (Zheng, 2003; Zuo and Dai, 2002), several worms that make great influence on the Internet, such as ''CodeRed'', ''Lion, Adore'', ''Nimda'' and ''Worm. KillMSBlast'' (EEye Digital Security, Code Red Worm; CERT, 2001; Zuo and Dai, 2002; Fearnow and Stearns, 2001; Mackie et al., Nimda worm analysis; Duba.net), may be programmed by security professionals in China.

The paper is structured as follows. Next section presents the definition, function structure and execution mechanism of Internet worms. Then, the scanning strategies of Internet worms are analyzed, followed by a discussion on the propagation models of Internet worms. Further, the techniques most frequently used to detect and prevent the attack of Internet worms are given. Furthermore, the future developments of Internet worms' researches are described. At last, the conclusion is given.

Function structure and work mechanism

Definition

The early main form of malicious code was the computer virus (Cohen, 1987, 1985). Spafford (1988) redefined the computer virus in order to distinguish the worm from the virus after the outbreak of ''Morris'' in 1988. He stated ''A virus is a piece of code that adds itself to other programs, including operating systems. It cannot run independently—it requires that its 'host' program be run to activate it.'' The Internet worm emphasizes its activity and independence. Kienzle and Elder (2003) gave the definition of Internet worm based on four aspects, namely, malicious code, network propagation, human intervention, and standalone or file-infecting. He stated ''A network worm is a piece of malicious code that propagates over a network without human assistance and can initiate actively attack independently or depending on file-sharing.'' Based on the propagation strategies, they grouped the worms into three categories: E-mail worms, windows file-sharing worms, and traditional worms. Zheng (2003) thought that the Internet worm had the properties of active attacking, concealing itself track, exploiting system vulnerability, blocking network traffic, decreasing system performance, repetition and devastation, etc. He also gave a definition accordingly: ''A network worm is a piece of independent program without the user intervention. It propagates itself through part or all of control privileges repeatedly gained by scanning vulnerabilities of computers on

network.'' This definition includes the latter two defined by Kienzle and Elder, excluding E-mail worms.

Schechter and Smith (2003) proposed a new type of network worm, the Access for Sale worm, at the ''WORM 2003'' conference in October, 2003. Besides the characteristics defined above, this type of worm has the property of identity authentication.

- Once released, it spreads from one system to another unaided by its author.
- It assigns a unique system identifier (USID) to each system it infects.
- Once inside a system, it creates a back door for remote access that opens only when presented with an access ticket containing its unique system identifier (USID).
- Only the author of the worm can generate valid access tickets from USIDs.
- The worm is matched with a mechanism by which the infection state and USID of a system may be obtained.

More information about the Access for Sale worm can be obtained in Schechter and Smith (2003).

According to the above analysis, we think that a worm is a kind of program or code that is intelligent and automatous, integrates hacker technologies with virus technologies, and can attack the hosts on network without human intervention. It scans and attacks hosts on a network with system vulnerabilities, and propagates itself from one host to another through the LAN or the Internet.

Function structure

Nazario et al. (2001) proposed a function structure framework of Internet worms. They thought that the core of any worm system consists of six components, i.e. reconnaissance capabilities, specific attack capabilities, a command interface, communication capabilities, intelligence capabilities, and unused attack capabilities. The framework mainly predicts the future research on network worms and is difficult to describe the current network worms. Based on the results of Spafford (1988), EEye Digital Security (Code Red Worm), CERT (2001), Zheng (2003), Zuo and Dai (2002), Fearnow and Stearns (2001), Mackie et al. (Nimda worm analysis), Duba.net and Nazario et al. (2001), we think that the function modules of worms can be classified to mainbody function modules and auxiliary function modules. The network worms with mainbody function modules can reproduce and propagate themselves, whereas other worms, which have both mainbody function modules and auxiliary function modules, have stronger survivability and devastation. The function structure is shown in Fig. 1.

Mainbody function module

The mainbody function module includes four submodules. The first is the information collection sub-module. This module specifies which search algorithm should be taken to collect information about the local or target network. The information includes user and password information, e-mail list, the hosts that are trusted or authorized by the local one, the topology of the network to which the local host belongs and boundary route information, etc. The information can be used alone or shared with the other individuals. The second is the probe module. This module scans and detects the vulnerabilities of the specified host, and determines which approach should be taken to attack and penetrate. The third is the attack submodule. This sub-module makes use of the holes gained by the probe sub-module to create a propagation path. In terms of attack approach, this sub-module should have good openness and extensibility. The last one is the self-propagating submodule. This sub-module uses various copies of worms and transfers these copies among different hosts. For example, the worm ''Nimda'' creates copies having different file formats and names (Mackie et al., Nimda worm analysis; CERT/CC, CERT Advisory). ''Worm.KillMSBlast'' propagates the function module using system programs, such as TFTP (Duba.net).

Table 1 lists some statistical data of the mainbody function modules of various well-known worms.

Auxiliary function module

Auxiliary function module is the accessory enhancing the mainbody function modules. It mainly

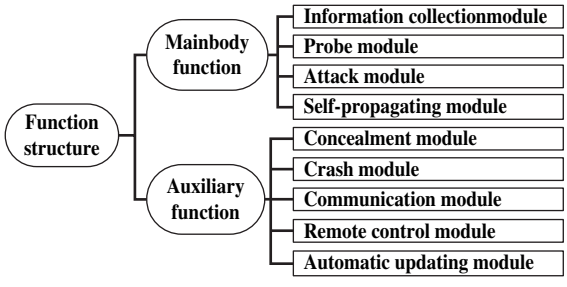

Figure 1 Function component of Internet worms.

Table 1 Main function component statistical information of some Internet worms

worm	Information collection	Probe (port)	Attack (system vulnerability)	Self-propagating (port)	Vulnerability exploited
Nimda	Yes	Yes (80, 139, 600)	Yes (IIS, Code Red II Sadmind backdoor)	Yes (80, 139, 600), E-mail and file-sharing	CA-2001-06
Code Red I, II	Yes	Yes (80)	Yes (IIS 4.0/5.0 Index Service)	Yes (80)	CA-2001-13, IN-2001-09
Adore	Yes	Yes (23, 53, 111, 515)	Yes (Bind, LPRng, Rpc.statd, wu-ftpd)	Yes (23, 53, 111, 515)	CA-2001-02, IN-2001-01
Sadmind/IIS	Yes	Yes (80, 111)	Yes (IIS, Solstice, Sadmind)	Yes (80, 111) 80: Windows 111: Unix	CA-2001-11, MS00-078
Lion	Yes	Yes (53)	Yes (BIND)	Yes (53)	CA-2001-02
Ramen	Yes	Yes (21, 111, 515)	Yes (wu-ftp, rpc.statd, LPRng)	Yes (21,111, 515) Worm copy: ramen.tgz	IN-2001-01
Cheese	Yes	Yes (10008)	Yes (Lion backdoor)	Yes (10008)	IN-2001-05
Digispid.B	Yes	Yes (1433)	Yes (Microsoft SQL Server)	Yes (1433)	IN-2002-04
Slapper	Yes	Yes (80, 443)	Yes (OpenSSL and Apache)	Yes (80)	CA-2002-27
MSSQL worm	Yes	Yes (1433)	Yes (Microsoft SQL Server)	Yes (1433)	CA-2003-04
W32.Blaster	Yes	Yes (135, 139, 445, 593)	Yes (Microsoft Dcom RPC)	Yes (135)	CA-2003-20

Notice: CA (CERT Advisory) and IN (CERT Incident Note) are alert information from CERT (Computer Emergency Response Team (CERT)).

includes five components: concealment module, crash module, communication module, remote control module, and automatic updating module. The concealment module comprises the concealment, transformation, encryption of the components of worm entity, as well as the concealment of process. This module mainly improves the survival capability of worms. The functions of the crash module include destroying or crashing infected hosts, breaching the normal network operation, planting backdoor in infected hosts, etc. The communication module enables the communication between worm and hacker or among worms, which is the mainstream of the future development of the worm. With the communication module, worms can share some information, which makes the programmer of the worm control its behaviour more effectively, and provides new communication channel for other modules. The remote control module is to regulate the worms' behavior, control infected hosts, and execute the instructions offered by its owner (programmer). The automatic update module allows the other modules to update its function momentarily and hence implements various further attack intentions.

Execution mechanism

According to the analysis of the function structure of Internet worms, we conclude that network worm is a kind of intelligent automatic attack program or code. It scans and detects the victim hosts over network having service holes, and once successful, will reproduce itself and create many copies which are then propagated from one host to another through the LAN or the Internet. The execution mechanism is shown in Fig. 2.

From the mainbody function modules of a network worm, we can conclude that the process of worm attack is composed of four stages: information collection, which mainly collects the information about the local and target host; probe, which detects the service holes of a specified target host; attack, which attacks the target host using the known vulnerabilities; self-propagation, which infects the target host.

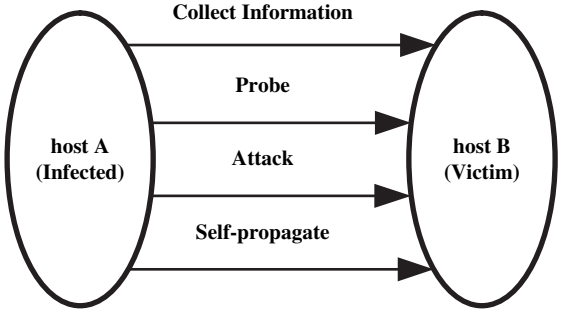

Figure 2 Execute mechanism of Internet worms.

Scanning strategy

Before initiating an attack, the worm should probe the system vulnerabilities of the target hosts. The ICMP Ping packet, TCP SYN, FIN, RST and ACK packets are all detected (Fyodor, 1997). A well-designed scanning strategy will accelerate the propagation of the worm. A worm with an ideal scanning strategy can find all the potential infectious computers over the Internet in the least time. Based on the different ways in which the worm selects the target address space, scanning strategies can be classified as follows: selective random scan, sequential scan, hit-list scan, routable scan, DNS scan, and divide-conquer scan.

Selective random scan

Instead of scanning the whole address space, worms randomly select part of the address set as the target address space, which has potential vulnerabilities. The selective address list is obtained from the whole or local route list. The unassigned addresses and reserved address block in Internet address space are excluded from the scan list. For example, there are 32 address blocks in Bogon list (Thomas, 2002). These address blocks identify those addresses that are not present in public network (Thomas, 2002). IPv4 address distribution map of IANA is a similar address list (Internet Protocol V4 Address Space). Selective random scan has several advantages. The algorithm is simple and easy to implement. If associated with the local preference strategy, the worm will propagate more effectively. However, the selective random scan has the limitation of easily blocking network traffic, which exposes the network worm early before it breaks out. ''CodeRed'' (EEye Digital Security, Code Red Worm), ''Slapper'' (Global Slapper Worm Information Center) and ''Slammer'' (Moore et al., 2003b) make use of this scanning strategy in order to spread rapidly.

Sequential scan

In the sequential scan, worms in infected host will select randomly an IP address of type C for propagation. According to the local preference strategy, it usually selects the IP addresses in the network to which it belongs. If the address the worm scans is A, the next IP address to be scanned will be $A + 1$, or $A - 1$. Once scanning a network with many susceptible hosts, the propagation will be more effective. The deficiency of this scan strategy is the repetition of scan, which may block the network traffic. Typically, ''W32.Blaster'' (EEye Digital Security, Blaster worm analysis) is a sequential scan worm.

Hit-list scan

The hit-list scan requires that the worm creates a target list which includes those hosts potentially infected before searching the susceptible hosts, and then tries to infect the computers listed (Staniford et al., 2002). The generation ways of hit-list include two types: (a) generating hit-list by scanning in miniature or sharing information of the Internet; (b) generating the whole list database by distribution scan. The ideal worm ''Flash'' is a hit-list scan worm based on IPV4 address space (Staniford et al., 2002; Zou et al., 2003a).

Routable scan

The routable scan (Zou et al., 2003b) is a kind of scan strategy in which network worms selectively scan IP address space based on the route information in a network. The worms using random scan usually detect the unassigned address spaces, most of which are not routable, as a result the propagation speed is affected heavily. If these network worms had known which IP addresses were routable, it would propagate more quickly and more effectively, and would escape from some anti-worm detecting systems.

The designers of the worms usually acquire the address prefixes from the major Internet backbones through the address spaces from BGP routing tables (CAIDA), and then verify the availability of BGB database. Routable scan increases greatly the propagation speed of worms. For example, as far as ''CodeRed'' is concerned, the infection probability of the worm using routable scan is 3.5 times than that of those using random scan (Zou et al., 2003b). But during the propagating processes, the worm must take a routing IP address database which instead results in the great big bulk.

DNS scan

The worms using DNS scan acquire a target address table from DNS server. The IP address table acquired with the DNS scan has the virtue of high usability and pertinence.

However, this scan has some problems. First it is difficult to acquire the whole address table from DNS records. Second, the address database the worms need to carry is so big that the propagations are very slow. Third, the number of addresses is limited to the number of the hosts with public domain names, for example, half of the hosts

infected by ''CodeRed'' are without DNS records (Moore et al., 2002).

Divide-conquer scan

The divide-conquer scan is a kind of scan strategy in which worms collaborate to search the susceptible hosts quickly. With divide-conquer scan, the worms send the part of the address database to each infected computer, who then scans the addresses acquired. For example, after host A infects host B, A sends part of the addresses it carried to B, and then B scans these addresses. A strategy to search a target list table using the divide-conquer scan is proposed in Kephart and White (1993).

The divide-conquer scan has the limitation of ''bad node'', that is, when propagating, if a node is turned off or broke down, all addresses sent to it would be lost. The earlier it takes place, the greater influence it makes. There are three ways to solve this problem: (1) create a target address list before sending address database; (2) control the propagation of worms by a counter—when a node is infected by a worm, the counter is increased by one, then tasks are assigned based on the value of the counter; and (3) determine randomly whether or not to pass the address database when worm is propagating.

Evaluation and discussion

There are four critical factors affecting the propagation speed of Internet worms: (a) selection of target address spaces; (b) whether or not to search susceptible host by multi-threads; (c) susceptible hosts list; and (d) the variety of propagation methods. The difference among various scan algorithms lies in the selection of target address spaces. The time in which Internet worms infect a host depends on the time it requires to search for a vulnerable computer. Therefore, designing algorithms for hunting vulnerable computers is the key to spread for Internet worms. Generally, the propagation speed using DNS scan is the slowest, while the speed of the selective random scan and the routable scan is quicker (Vogt, 2003). As far as Hit-list scan is concerned, when the size of the list exceeds 1 MB, the propagation speed will be slower than the routable scan, and when the size is over 6 MB, the speed will be even slower than that of random scan (Vogt, 2003). Therefore the address database the worms carry should not be too large. For divide-conquer scan, it is difficult to seek an effective and easily implemented algorithm. Currently, the propagation using routable scan and then random scan is the most optimal propagation method.

Propagation model

An accurate Internet worm propagation model can have an insight into worm behavior, identify the weakness in the worm spreading chain and provide accurate prediction for the purpose of damage assessment for a new worm threat. As for the study of the malicious logic propagation models, there are many virus propagation models (Anderson and May, 1991; Bailey, 1975), but few worm propagation models. The propagation models of infectious diseases are applicable to the propagation for worms (Bailey, 1975; Andersson and Britton, 2000; Frauenthal, 1980; Allen and Burgin, 2000; Chen et al., 2003). Infectious diseases models include Simple Epidemic model (Andersson and Britton, 2000), Kermack—Mckendrick model (Frauenthal, 1980), SIS (Susceptible → Infectious → Susceptible) model (Allen and Burgin, 2000), Two-Factor model (Zou et al., 2002) and the WAW (Worm-Anti-Worm) model proposed in this paper.

Simple Epidemic model

In Simple Epidemic model, SEM (Andersson and Britton, 2000), each host is in one of the two states: susceptible or infectious. The model also assumes that once infected by a virus, the host remains in the infectious state forever. Thus the transition procedure is: susceptible → infected (Frauenthal, 1980). The mathematical expression for the infected host can be described with a differential equation (Zou et al., 2003a):

$$dI(t)/dt = \beta I(t)[N - I(t)] \quad (1)$$

where $I(t)$ denotes the number of infectious hosts at time t, N the number of hosts in system, and β is the rate of infection in epidemiology studies. At $t = 0$, $I(0)$ hosts are all infectious and the other $N-I(0)$ are all susceptible.

Let $a(t) = I(t)/N$, dividing both sides of Eq. (1) by N^2 we have

$$da(t)/dt = Ka(t)[1 - a(t)] \quad (\text{where } K = \beta N) \quad (2)$$

Assume $N = 10\,000\,000$, the rate of infection $\beta = 1/10\,000\,000$, such that $K = \beta N = 1$, the number of infected hosts $I(0) = 3$. The simulation is shown in Fig. 3, where x-coordinate is the propagation time delay and the y-coordinate the infected percentage of the whole Internet.

SEM model can describe the propagation status in initial stages, but is difficult to match the propagation status later.

Kermack—Mckendrick model

Unlike the SEM model, the host in Kermack—McKendrick model (KM model) maintains one of three states: susceptible, infectious or removed (Frauenthal, 1980). The KM model is expressed by a differential equation as follows:

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

where $I(t)$ denotes the number of infectious hosts at time t, $R(t)$ the number of removed hosts from the infected hosts at time t, and $J(t)$ the number of infected hosts including the hosts that are still infectious and those being immune from the infectious, that is $J(t) = I(t) + R(t)$, β the rate of infection, γ the rate of recovery from the infected host, $S(t)$ the number of vulnerable hosts at time t, and N is the number of node hosts in system.

As far as KM model is concerned, the immunity of an infected host means removal of the hosts from the whole system. Accordingly, the number of hosts reduces from N to $N-1$.The propagation trend of KM model is shown in Fig. 4, in which $N = 10\,000$, $\beta = 1/10\,000\,000$. If $J(0) = 3$, then $\gamma = 0.001$. At last, the whole number of hosts and infectious hosts in system will reduce to 0.

KM model considers the immune states of infectious hosts based on SEM model and describes worm propagation more precisely. However, the KM model excludes the situation where susceptible and infected hosts are patched to resist the worm. In addition, it is not appropriate to assume the rate of infection to be constant.

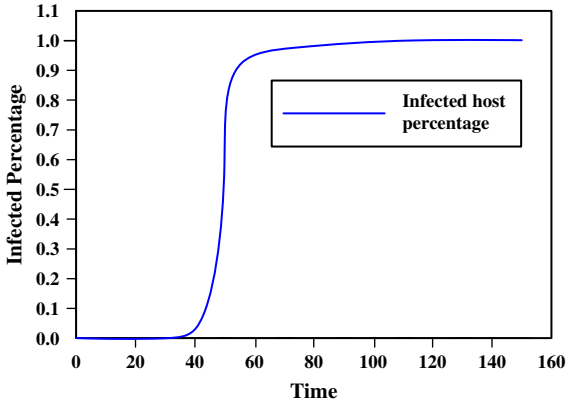

Figure 3 Internet worm propagation trend in SEM model.

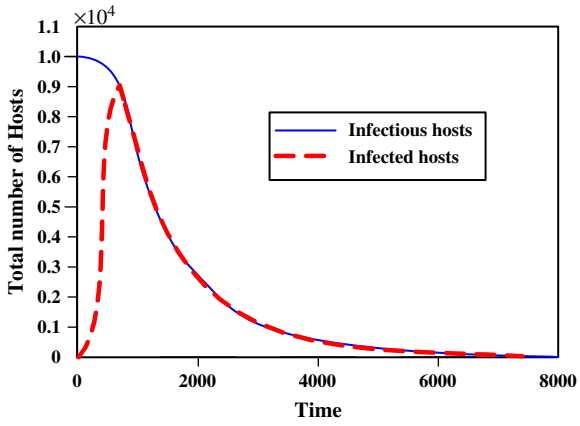

Figure 4 Internet worm propagation trend in KM model.

Susceptible—Infectious—Susceptible model

Unlike the KM model, the Susceptible—Infectious—Susceptible (SIS) model assumes every host has the same possibility of being infected repeatedly (i.e., recovered host has the same possibility of being infected as susceptible host). However, the model doesn't take account of the situation that the infected hosts are patched or updated to be immune from the worms. The SIS model is not applicable to describe the infection of Internet worms, the mathematical expression for the infected host can be described with a differential equation (Allen and Burgin, 2000):

$$dI(t)/dt = \beta I(t)[N - I(t)] - \gamma I(t) \quad (4)$$

where $I(t)$ denotes the number of infectious hosts at time t, N the number of hosts in system, β the rate of infection in epidemiology studies, and γ is the rate of recovery.

If $\beta > \gamma$, then the rate of infection in network is quite close to $1-\gamma/\beta$, and the number of infected hosts and that of all hosts maintain certain stable relationship at last, the ratio is close to $1-\gamma/\beta$. If $\beta < \gamma$, then the worms stay latent states.

Two-Factor model

Two-Factor model considers more external factors and anti-worm measures than the models above (Zou et al., 2002). One factor is the dynamic countermeasures taken by ISPs and users; the other is the slowed down worm infection rate because rampant propagation of worm causes congestion and troubles to some routers. The parameters of $\beta(t)$, $R(t)$ and $Q(t)$ dynamically

change with time t, the mathematic expressions reflecting their dynamic change are as follows:

$$\begin{cases} dR(t)/dt = \gamma I(t) \\ dQ(t)/dt = \mu S(t) J(t) \\ \beta(t) = \beta_0 [1 - I(t)/N]^{\eta} \\ N = S(t) + I(t) + R(t) + Q(t) \\ dS(t)/dt = -\beta(t) S(t) I(t) - dQ(t)/dt \end{cases} \quad (5)$$

where $\beta(t)$ denotes the rate of infection at time t, $I(t)$ the number of the infectious hosts at time t, $R(t)$ the number of the hosts that are immune after being infected at time t, $Q(t)$ the number of the hosts that are immune before being infected at time t, $J(t)$ the number of infected hosts at time t, $J(t) = S(t) + R(t)$, $S(t)$ the number of infectious hosts at time t, and γ, μ and β_0 are constants. From Eq. (5), we can get the relationship between $I(t)$ and t, and this is Two-Factor model expression (Zou et al., 2002).

$$dI(t)/dt = \beta(t)[N - R(t) - I(t) - Q(t)]I(t) - dR(t)/dt \quad (6)$$

The propagation trend of Two-Factor model is shown in Fig. 5 in which $N = 1\,000\,000$, $I_0 = 1$, $\eta = 3$, $\gamma = 0.05$, $\mu = 0.06/N$, $\beta_0 = 0.8/N$. With the $Q(t)$ increasing, $I(t)$ tends to 0.

The Two-Factor propagation model is the extension and supplement of SEM and KM, and is more suitable to propagation states of Internet worms. However, this model still does not consider that the infected hosts are patched or updated to confront worms. Moreover, the condition of the worm against the worm complicates the worm propagation model.

Worm-Anti-Worm model

This model considers two types of worms: a malicious worm A and an oppositional one B. We divide the propagation process into two stages: when B is absent, the propagation of A is subject to the Two-Factor model; when B is present, there are four potential cases: (1) B detects and cleans A, and patches the hosts infected by A; (2) B only detects and cleans A; (3) B patches all susceptible hosts; and (4) B patches all susceptible hosts, and detects and cleans A. In the first two cases, B only searches the infected hosts, while in the latter two B searches all susceptible hosts. The first situation follows the KM model, where the immunity speed of the susceptible hosts is higher than that when B is absent. The second situation is typically an SIS model. The last two situations supplement the Two-Factor model in the aspects of countermeasures, and principally influence the

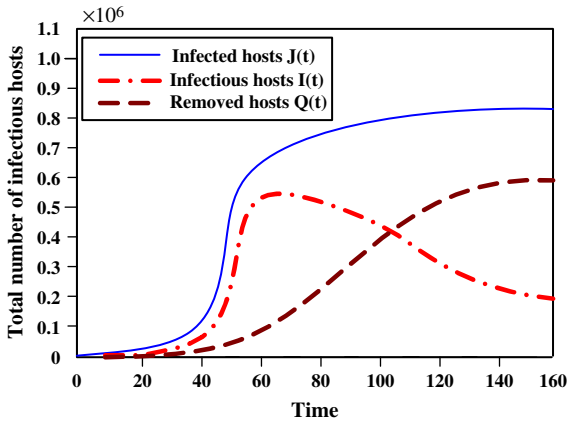

Figure 5 Internet worm propagation trend in Two-Factor model.

cleaning speed of worm A in the subsequent propagation stage. This paper discusses the propagation model of A in the fourth situation. Based on the Two-Factor model, the change in the number of susceptible hosts $S(t)$ from time t to time $t + \Delta t$ follows:

$$dS(t)/dt = -\beta(t) S(t) I(t) - dQ(t)/dt \quad (7)$$

where $S(t)$ is the number of all susceptible hosts at time t for worm B, and there are only two states in the system: susceptible and infectious. The propagation of B follows the SEM model. The differential equation expressing the infectious hosts is as follows:

$$dR_B(t)/dt = \beta R_B(t)[S(t) - R_B(t)] \quad (8)$$

where $R_B(t)$ is the host that B repairs at time t. According to Eqs. (5) and (8), the mathematic expression for Worm-Anti-Worm model:

$$\begin{cases} dR(t)/dt = \gamma I(t) + dR_B(t)/dt \\ dQ(t)/dt = \mu S(t) J(t) \\ \beta(t) = \beta_0 [1 - I(t)/N]^{\eta} \\ N = S(t) + I(t) + R(t) + Q(t) \\ dS(t)/dt = -\beta(t) S(t) I(t) - dQ(t)/dt - dR_B(t)/dt \\ dR_B(t)/dt = \beta_1 R_B(t)[S(t) - R_B(t)] \end{cases} \quad (9)$$

The propagation trend of the Worm-Anti-Worm model is shown in Fig. 6, where $N = 1\,000\,000$, $I_0 = 1$, $\eta = 3$, $\gamma = 0.05$, $\mu = 0.06/N$, $\beta_0 = \beta_1 = 0.8/N$, the time difference between the emergence of A and B $\Delta t = 100$. From this figure, we can see that the worm A vanished quickly.

The Worm-Anti-Worm model takes the existence of the antagonistic worm into account and more precisely predicts the propagation trend in the later part. However, this model doesn't consider the relationship between the propagation of

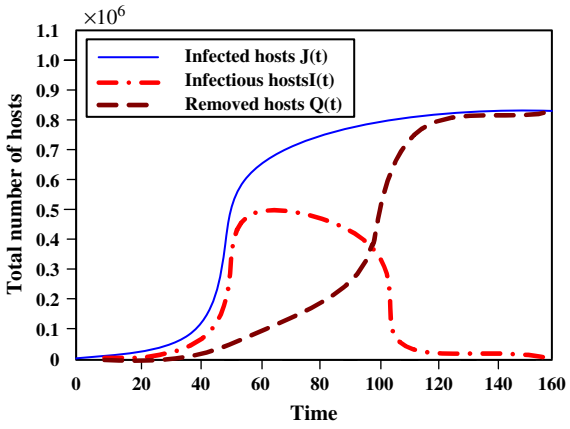

Figure 6 Internet worm propagation trend in WAW model.

the antagonistic worm itself and the other limiting factors, as well as the states of the antagonistic worm after it enters the susceptible hosts.

Detection and defense

Internet worms have become a leading menace to the Internet system. Because of the complexity and uncertainty of worm activities, the defense against worms needs to integrate various technologies, including monitoring and early warning of the worm, blocking the worm, repairing the system hole automatically, propagation restraint and emergency response on Internet worms, etc. This section summarizes the main detection and defense technologies in recent years.

GrIDS and Netlike relevant analysis

The well-known GrIDS (Cheung et al., The design of GrIDS) is designed to detect the large-scale network attack and the automated invasion on the network. The system collects the network activity data from Internet, and uses the data to construct the network activity graph to describe the causal relation in the network activity structure through the pattern database defined in advance. By establishing and analyzing an activity graph among nodes, it detects whether the worm does exist through carrying on the match with the predefinition behavior pattern graph. Currently GrIDS is an effective tool to defend the distribution network worm invasion. However, GrIDS still has several deficiencies. Firstly, the detection agent of GrIDS does not carry on context-based relevant analysis on the package information which is transmitted over network, does not make full use of much more, even effective information, and only makes simply event-based connection analysis. Secondly, GrIDS does not do effective analysis on the target addresses and the target service in the TCP connection, yet this analysis is the important basis for determining unknown worm intrusion. Lastly, after GrIDS detecting a network worm, GrIDS still does not establish any response mechanism or provide any interaction with the interior detection agents and exterior firewall, therefore, it cannot give an effective early warning or defense mechanism.

In view of the weaknesses mentioned above, we have designed a new technique based on the network related analysis to analyze and warn the worm attack. The technique adopts a distribute system structure, makes full use of the information provided by various detection agents and is based on the methods of data mining and abnormal detection, and through making relevant analysis about data on various detection points, it implements early warning to predict network worm in a large-scale network environment.

PLD system

Lockwood et al. (2003) in Applied Research Laboratory, Washington University, proposed a defense system against Internet worms using Programmable Logic Devices (PLDs). The system was comprised of three interconnected components: a Data Enabling Device (DED), a Content Matching Server (CMS), and a Regional Transaction Processor (RTP). These elements work together to provide network wide protection. The DED is responsible for capturing each packet of data as it is in and out of the network, scanning and matching it based on fixed strings or regular expressions CMS provides, and then forwarding the results to RTP. The CMS reads stored information on worm from a backstage MYSQL database, and compiles and integrates it into fixed-string or regular expression used by DED. The RTP bases the matching content to consult a database to determine the action that the DED should take. Whenever a new worm outbreak occurs, an administrator adds the signature of the worm to the database table on CMS. The DED then scans the live Internet traffic for the targeted signature. Whenever the matching content is found, the DED requests the RTP to either block the traffic or allow it to pass.

The system has several advantages. Firstly, the core of the DED is the high-speed hardware FPX

(Field-programmable Port Extended) (Lockwood et al., 2001). The FPX can process data at a rate of 2.4 Gbps, so the system is suitable for the worm detection in a large-scale, high-speed network environment. Secondly, the high-speed hardware FPX can implement parallel more easily than software system do.

But this method cannot warn the worm in advance and can't detect unknown worms. In addition, due to using characteristic matching technology, there is a certain amount of error when using this method.

HoneyPot

The HoneyPot was originally used to prevent network hacker attack (Honeypot technology). Revirt is a kind of HoneyPot system that detects a network attack and network abnormal activities (George et al., 2002). Spitzner (2002) was the first to adopt HoneyPot to prevent malicious codes attack. The literature (Provos, A virtual Honeypot framework) proposed a prevention framework using virtual HoneyPot to detect and block network worm attacks. We may deploy a number of virtual HoneyPots at boundary gateways or vulnerable places. These virtual honeyPots can share the captured information and use an automated NIDS signature generator to generate a matching database. When network worms use some scan strategy to scan the address space of those hosts' existing holes, the HoneyPots will capture the information about worm scanning and attack, and then depend on signature matching to determine whether an attack takes place (Provos, A virtual Honeypot framework). In addition, HoneyPot can interrupt the attack of network worms. Oudot (2003) used the HoneyPot to detect and prevent the W32.Blaster successfully.

HoneyPot has some advantages: (1) HoneyPot can transfer the worms' attack targets, and decrease the attack effect; (2) HoneyPot provides much information for network security professionals to research the function mechanism of the worm, track the source of the attack and predict the attack targets, etc. and (3) HoneyPot has good concealment because the network worms lack the ability to judge the usability of the targeted system.

The deficiencies of the HoneyPot are: (1) whether or not the network worms are tricked by the HoneyPot depends on a number of factors, including the name of the HoneyPot, the position where the HoneyPot is deployed and the reliability of the HoneyPot itself; (2) HoneyPot can detect worms using various scan activities, such as random scan and sequential scan, but it is difficult for those using routable scan and passive propagation; and (3) HoneyPot rarely produces good results in the early propagation stage.

Benign worms restrain malicious worms

The earliest worm was introduced in order for science aided computation and performance testing of large-scale networks (Shoch et al., 1982). The worm itself has the characteristic of distributed computation. Therefore the benign worm can be used to prevent the malicious worm. The benign worm must firstly be highly controllable, and then avoid increasing the network payload as far as possible. The benign worms can take several propagation methods: (1) make use of the backdoor that the malicious worm left behind; (2) use the holes set by the attacks of malicious worm; (3) employ other system holes unopened; and (4) use the authorization of the attacked hosts. The benign worm can decrease the number of susceptible hosts in the system effectively, clear up malicious worms and repair the system holes. The worm "Cheese" (Barber, Cheese worm; CERT/CC, CERT® Incident Note) can control infected hosts by the backdoor that the worm "Lion" (Zuo and Dai, 2002; Kasarda, The Lion worm) leaves, and then eliminates this backdoor and repairs these holes. The worm "CodeGreen" (HexXer, CodeGreen source code) and "CRClean" (Kern, CRClean source code) against "CodeRed" had even been published before. But they were not applicable to the real network. "Worm.KillMSBlast" uses the hole exploited by "W32.Blaster" against "W32.Blaster". These are classical instances of the worm against worm. However, "Cheese", "CodeGreen", "CRClean" and "Worm.KillMSBlast" are not benign, because they seriously affect the network payload.

The advantages of benign worms are as follows. Firstly, a benign worm is transparent for the users. It is not necessary for the benign worm to conceal modules. The benign worms can acquire mainbody programs, data and propagation targets through central controlling. Secondly, the benign worm propagates slowly in time-sharing and section-sharing in order not to take the excessive width and resources as far as possible. Thirdly, the same benign worm can be used to carry out different tasks, only requiring downloading different task modules from the control center, such as performing distributed computations and collecting

network data, and then submitting the results to the center.

The future research focuses for network will be the benign worm with the key factor being controllability. Therefore, designers must put more effort into understanding unpredictable factors for benign worms.

CCDC framework

Since network worms have biological virus characteristic, the American security experts proposed to establish the Cyber Center for Disease Control Framework, CCDC Framework, to defend against network worms' attack (Weaver et al., 2003). The CCDC implements the following functions: (1) discerning the outbreak of the worm; (2) analysis of the worm sample characteristic; (3) worm infection resistance; (4) prediction of new infection methods; (5) study of worm resistance tools in advance; and (6) threat resistance to future worms. CCDC realizes early warning, defends and blocks large-scale networm invasion. But CCDC also has some deficiencies. Firstly, CCDC is a large-scale defense system and so the running cost must be considered. Secondly, because of the openness of the system, the security of the CCDC is another considerable question. Lastly, in the CCDC defense system, the attackers can monitor the whole attack process, and understand the function mechanism of CCDC, which may result in the design of a worm that breaks through the CCDC defense system (Weaver et al., 2003).

Other methods

In addition to the above technologies, there are many other worm defense technologies. For example, deploying network or firewall software and closing ports other than the normal service ports will cut off the transportation passages and communication channels of the worms. The others include filtering the messages containing some worms' characteristic and preventing the infected hosts from accessing the protected network, etc. The prevailing approach to restrain worm propagation is to close and filter the messages that contain some worm's characteristics at routers. Moreover, Zou et al. (2003c) proposed to predict the worm propagation by monitoring some address space traffic and then take more effective measures to resist the worm attack. The tool LaBrea (Liston, Welcome to my tarpit), designed by Liston can decrease the propagation speed by blocking the uninfected hosts from TCP connection to the infected machines for a long time (Balasubramaniyan et al., 1998; Porras and Neumann, 1997).

Research trend

By analyzing the function structure and the execution mechanism of Internet worms, we think there are several meaningful research directions on Internet worm implementation.

The first is to synthesize the attack technologies of the virus and Trojans. Having broken through the system, more and more worms continue to attack the file system, which results in the diversity problems of the propagation. Early virus distortion technology and automatic generation technology will also be integrated to compile the worm codes, resulting in the worm polymorphism. The worms also use the concealment technology that Trojan takes, including the individual concealment, the process concealment, the spot recovery, etc. The kernel level hacker attack and defense technology will also be integrated into the worm function to hide the worm track.

The second is the function dynamic updating technology. The worms can dynamically update all function modules, and thus acquire stronger survivability and attack capability. This technology enables the designers of worms be able to update the worms' function momentarily, accordingly realizing different attack intention.

The third technology is the intelligent detection technology. This technology uses the existing functions of the network, with the aid of the search engine, to obtain useful information, including the address list of active servers and relevant information of certain user. Worms depending on the search engine to acquire information are more effective than the ones to carry IP addresses.

The fourth is the cross-platform technology. Through this technology, the worm carries codes that can run on different platforms, resulting in cross-platform propagation. We believe that mobile phone worms and electrical appliances worms will appear in near future.

The last is distributed cooperation computation technology. For the distributed computation worm, its data and code are stored in different places. When the worm begins an attack, it gains attack information from a control center. At the same time, attack codes use certain algorithms to search and reproduce data at various points. Different function modules distribute in different hosts and cooperate to generate stronger

concealment and attack capability. "Cactus Worm" (Allen et al., The cactus worm) and "Smart Worm" (Ellis, 2002) are two well-known distributed worms.

Conclusion

In this paper we first presented the concepts and research situations of Internet worms, as well as function components and execution mechanisms. Then the scanning strategies and propagation models were discussed, and the critical techniques of Internet worm prevention were given. As far as the development of the worm is concerned, the hot issues about networms are as follows: (1) quick scanning strategy and propagation mechanism; (2) the propagation model and simulation test; (3) mathematical computation model research; (4) research on early warning and block technology; (5) hide mechanism and activation mechanism; and (6) tracing and evidence collection of Internet worms.

The detection and prevention of network worms is a long-term process. This is mainly attributed to two reasons: (1) the diversity of the worm types is complex and they change repeatedly; and (2) it is difficult to accurately foresee new network worms. Therefore, we must not only grasp the current execute mechanism of networms, but also strengthen the research on the development trends and actually prevent incidents before they break out.

References

Allen LJ, Burgin AM. Comparison of deterministic and stochastic SIS and SIR models in discrete time. Mathematical Biosciences 2000;163:1–33.

Allen G, Angulo D, Foster I. The cactus worm: experiments with dynamic resource discovery and allocation in a grid environment. <http://xxx.lanl.gov/pdf/cs.DC/0108001>.

Anderson RM, May RM. Infectious diseases of humans: dynamics and control. Oxford: Oxford University Press; 1991.

Andersson H, Britton T. Stochastic epidemic models and their statistical analysis. New York: Springer-Verlag; 2000.

Arnold B, Chess D, Morar J, Segal A, Swimmer M. An environment for controlled worm replication and analysis. Published at the Virus Bulletin; September 2000. p. 1–20.

Bailey NT. The mathematical theory of infectious diseases and its applications. New York: Hafner Press; 1975.

Balasubramaniyan JS, Garcia-Fernandez JO, Isacoff D, Spafford E, Zamboni D. An architecture for intrusion detection using autonomous agents. Technical Report 98/05, Purdue University; 1998.

Barber B. Cheese worm: pros and cons of a friendly worm. <http://rr.sans.org/malicious/cheese.php>.

CAIDA. IPv4 BGP geopolitical analysis. <http://www.caida.org/analysis/geopolitical/bgp2country/>.

CERT. Code Red II: another worm exploiting buffer overflow in IIS indexing service DLL. <http://www.cert.org/incident_notes/in-2001-09.html>; 2001.

CERT/CC. CERT Advisory CA-2001-26 Nimda worm. <http://www.cert.org/advisories/CA-2001-26.html>.

CERT/CC. CERT® Incident Note IN-2001-05. <http://www.cert.org/incident_notes/IN-2001-05.html>.

Chen Z, Gao L, Kwiat K. Modeling the spread of active worms. In: IEEE INFOCOM 2003. IEEE; April 2003.

Cheung S, Hoagland J, Levitt K, Rowe J, Staniford C, Yip R, et al. The design of GrIDS: a graph-based intrusion detection system. Technical report CSE-99-2. U.C. Davis Computer Science Department. <http://citeseer.nj.nec.com/cheung99design.html>.

Cohen F. Computer viruses. Ph.D. thesis, University of Southern California; 1985. p. 1–5.

Cohen F. Computer viruses—theory and experiments. In: DOD/NBS 7th conference on computer security, originally appearing in IFIP-sec 84 [also appearing in Computers and Security 1987;6:22–35].

Computer Emergency Response Team (CERT). <http://www.cert.org/advisories/>.

EEye Digital Security. Blaster worm analysis. <http://www.eeye.com/html/Research/Advisories/AL20030811.html>.

EEye Digital Security. Code Red worm. <http://www.eeye.com/html/research/advisories/al20010717.html>.

Ellis D. A potency relation for worms and next-generation attack tools. MITRE Technical report; 12 March 2002.

Fearnow M, Stearns W. Adore worm. <http://www.sans.org/y2k/adore.htm>; April 2001.

Frauenthal JC. Mathematical modeling in epidemiology. New York: Springer-Verlag; 1980.

Fyodor. The Art of port scanning. Phrack Magazine. September 1997;7(51):11–7.

George WD, Samuel TK, Sukru C, Murtaza B, Peter MC. ReVirt: Enabling intrusion analysis through virtual-machine logging and replay, Proceedings of the 2002 symposium on operating systems design and implementation; December 2002.

Global Slapper Worm Information Center. <http://www.f-secure.com/slapper/>.

HexXer H. CodeGreen source code. <http://www.incidents.org/archives/intrusions/msg00808.html>.

Honeypot technology. <http://www.xfocus.net/articles/200103/121.html>.
<http://www.duba.net/c/2003/08/21/90290.shtml>.

Internet protocol V4 address space. <http://www.iana.org/assignments/ipv4-address-space/>.

Kasarda A. The Lion worm: king of the jungle? <http://rr.sans.org/malicious/lion.php>.

Kephart JO, White SR. Measuring and modeling computer virus prevalence. In: Proceedings of the IEEE symposimum on security and privacy; 1993. p. 2–15.

Kephart JO, Chess DM, White SR. Computers and epidemiology. IEEE Spectrum 1993;30(5):20–6.

Kern M. CRClean source code. <http://archives.neohapsis.com/archives/vulndev/2001-q3/0577.html>.

Kienzle DM, Elder MC. Recent worms: a survey and trends. WORM'03; October 2003.

Liston T. Welcome to my tarpit – the tactical and strategic use of LaBrea. <http://www.hack.buster.net>.

Lockwood JW, Naufel N, Turner JS, Taylor DE. Reprogrammable network packet processing on the Field Programmable Port Extender (FPX). In: ACM international symposium on field programmable gate arrays (FPGA); February 2001. p. 87–93. Monterey, CA, USA.

Lockwood JW, Moscola J, Kulig M, Reddick D. Tim Brooks. Internet worm and virus protection in dynamically reconfigurable

hardware. Military and Aerospace Programmable Logic Device (MAPLD), Washington DC; 2003. Paper E10, September 9–11, 2003.
Mackie A, Roculan J, Russell R, Velzen MV. Nimda worm analysis. <http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf>.
Moore D, Shannon C, Claffy K. Code Red: a case study on the spread and victims of an Internet worm. In: Proceeding Internet measurement workshop; November 2002. p. 273–84.
Moore D, Shannon C, Voelker G, Savage S. Internet quarantine: requirements for containing self-propagating code, Proceedings of the 2003 IEEE Infocom conference April 2003. San Francisco, CA. <http://www-cse.ucsd.edu/users/savage/papers/Infocom03.pdf>.
Moore D, Paxson V, Savage S, Shannon C, Staniford S, Weaver N. Inside the slammer worm. IEEE Magazine of Security and Privacy July/August 2003;33–9.
Nazario J, Anderson J, Wash R, Connelly C. The future of Internet worms. Presented at the Blackhat Briefings. <http://www.crimelabs.net/docs/worm.html>; July 2001.
Oudot L. Fighting worms with Honeypots:Honeyd vs Msblast.exe. Honeypots mailinglist; August 2003. <http://lists.insecure.org/lists/honeypots/2003/Jul-Sep/0071.html>.
Porras PA, Neumann PG. Emerald: event monitoring enabling responses to anomalous live disturbances. In: Proceedings of the 20th national information systems security conference; October 1997. p. 353–65.
Provos N. A virtual Honeypot framework. CITI Technical report 03-1. <http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf>.
Schechter SE, Smith MD. Access For Sale: a new class of worm. In: Proceedings of the 2003 ACM workshop on Rapid Malcode; 2003. p. 138–47. Washington, DC.
Shoch, John F, Jon AH. The worm programs early experience with a distributed computation. Communications of the ACM 1982;25(3):172–80.
Song D, Malan R, Stone R. A snapshot of global Internet worm activity. Arbor Networks, Technical report; November 2001. <http://www.first.org/events/progconf/2002/d5-02-song-slides.pdf>.
Spafford EH. The Internet worm program: an analysis. Technical report CSD-TR-823, Department of Computer Science, Purdue University; 1988. p. 1–29.
Spitzner L. Honeypots: tracking hackers. Addison Wesley Professional; September 2002.
Staniford S, Ellis D, Weaver N. The worm information center. <http://www.networm.org/>.
Staniford S, Paxson V, Weaver N. How to own the Internet in your spare time. In: 11th Usenix security symposium; August 2002. San Francisco. <http://www.icir.org/vern/papers/cdc-usenix-sec02/cdc.pdf>.
Steve W. Open problems in computer virus research. <http://www.research.ibm.com/antivirus/SciPapers/White/Problems/Problems.html>; October 1998.
Thomas R. Bogon list v1.5. <http://www.cymru.com/Documents/bogon-list.html>; August 2002.
Vogt T. Simulating and optimizing worm propagation algorithms. <http://web.lemuria.org/security/WormPropagation.pdf>; September 2003.
Weaver N. Warhol worms: the potential for very fast Internet plagues. <http://www.cs.berkeley.edu/tildenweaver/warhol.html>.
Weaver N. Potential strategies for high speed active worms. <http://www.cs.berkeley.edu/~nweaver/worms.pdf>; March 2002.
Weaver N, Paxson V, Staniford S, Cunningham RLarge scale malicious code: a research agenda; 2003. p. 11–16.
Yang S, Relations M. NSF awards $5.46 million to UC Berkeley and USC to build test bed for cyber war games. <http://www.berkeley.edu/news/media/releases/2003/10/15_testbed.shtml>.
Zheng H. Internet worm research [for the degree of PhD]. Information Technologies & Science College, Nankai University, Tianjin, P.R. China; 2003. p. 12–15.
Zuo XD, Dai YX. Analysis on Lion worm and some discussing about it. Computer Engineering 2002;28(1):16–7.
Zou CC, Gong W, Towsley D. Code Red worm propagation modeling and analysis, 9th ACM Symposium on computer and communication security 2002; Washington, DC.
Zou CC, Gong W, Towsley D. On the performance of Internet worm scanning strategies. Mass ECE Technical report TR-03-CSE-07; November 2003.
Zou CC, Towsley D, Gong W, Cai S. Routing worm: a fast, selective attack worm based on IP address information. Umass ECE Technical report TR-03-CSE-06; November 2003.
Zou CC, Gao L, Gong W, Towsley D. Monitoring and early warning for Internet worms. Umass ECE Technical report TR-CSE-03-01; 2003.

WeiPing Wen was born in 1976. He is a Ph.D. student of the Engineering Research Center for Information Security Technology, the Institute of Software, the Chinese Academy of Sciences. His research interests are theory and technology of network and information security and research on malicious code.

Sihan Qing was born in 1939. He is a chief researcher in Engineering Research Center for Information Security Technology, Institute of Software, Chinese Academy of Sciences and a supervisor of Ph.D. candidates. His research interests include theory and technology of network and information security, secure operating system, design and analysis of cryptographic protocols, intrusion detection system, etc.