ELSEVIER

Computer Networks and ISDN Systems 27 (1995) 1447-1456

COMPUTER NETWORKS and ISDN SYSTEMS

Quantitative risk assessment of computer virus attacks on computer networks

B.C. Soh*, T.S. Dillon, P. County

Computer Networks Laboratory, Applied Computing Research Institute, and Department of Computer Science & Computer Engineering La Trobe University, Melbourne, Australia 3083

Accepted 19 August 1994

Abstract

This paper discusses the various types of malicious software, particularly computer viruses, which threaten computer network dependability, including such attributes as reliability, availability, safety and security of computer systems. Quantitative risk assessment of computer virus attacks on computer networks is investigated. To this end, an analytical model to study computer virus propagation in a typical network is developed and the results are presented. The model developed in this paper theoretically supports what is commonly performed in network management, where particular network resources are to be protected or preserved under hostile conditions. Some strategic controls are discussed.

Keywords: Computer viruses; Computer network dependability; Analytical model; Risk assessment

1. Introduction

In the past few years, several breaches of computer network security arising from computer viruses have been experienced. This has led to considerable interest in the development of strategies that would contain computer viruses [12], protect against possible computer virus attack [8], detect computer viruses [4], assist recovery from computer virus attack and help in the design of computer-virus-resistant operating systems [8]. Quantitative risk assessment of a computer virus attack on computer networks has not been investigated in any detail, to the best of our knowledge.

Any managerial planning by an organization for computer network dependability, particularly with regard to security, should start with an assessment of the risks faced and the losses that would accrue to the organization in the event of a computer virus attack. As computer viruses often pose very real threats to computer network dependability, and as no 100% effective security measures against the threat are available (except with infinite cost), the risk of attack should be quantified so that counter measures can be evaluated.

In Section 2, we describe various different types of malicious software, some of which are frequently misunderstood (in particular *worms* are often mistaken for *viruses*, and vice versa). Section 3 describes and compares the properties of computer viruses and worms; computer viruses are then the focus of study in Sections 4 to 11. Section 4 describes how computer systems can be attacked by computer viruses, while Section 5 gives the definition of computer virus attack risk used in this paper. In Section 6 we give the notation to be used in a mathematical model developed in Section 7 for investigating computer virus er-

* Corresponding author.

SSDI 0169-7552(94)00084-0

ror propagation and predicting the probability of computer virus error propagation through an entire computer network. Based on the model developed, a solution method is presented in Section 8. In Section 9, we describe some sensitivity studies, and present the results of these studies. In Section 10, we quantify the risk assessment by defining a measure called *criticality*, while in Section 11 we discuss the strategic controls, taking criticality into account. Section 12 concludes this paper with discussion of directions for future work.

2. Malicious software terminology

The following types of malicious software have been understood differently by different people. In this paper we shall closely follow the definitions below, which were proposed by the Sub-committee on Computer Ethics of the IEEE Computer Society Committee on Public Policy [10]:

- a 'trap door' is something which "provides a hidden software (or hardware) mechanism that permits computer system protection to be circumvented";
- a 'time bomb' is a "trap door that is activated by a particular set of circumstances, usually occurring over time";
- a 'Trojan horse' is a "program with an apparently or actually useful function, but that also performs an unexpected (deleterious) function";
- a 'worm' is a "program that replicates itself throughout a computer network through its own exclusive efforts. In the process, it overlays or erases other programs or data, making them useless";
- a 'computer virus-A' is a "Trojan horse program which is allowed to spread to (and is therefore said to infect) another computer";
- a 'computer virus-B' is a "Trojan horse program which can modify the executable code of another program and add the malicious instructions that now cause the other program to become a Trojan horse". (Also see [3].)

It is interesting to note that virus-B is more insidious than virus-A since the former can modify executable code of another program. In this paper the term 'computer virus' can mean either a virus-A or a virus-B.

3. Properties of computer viruses and worms

The first report of computer viruses was in 1981. Adleman [1] is credited with coining the term "computer virus", while Cohen [3,4] is credited with doing the first serious research in the area. Not all computer viruses are malicious. A computer virus can be used to compress or decompress programs or files so that less memory is occupied, and at the same time to spread the compression function to other programs [3,4]. As mentioned above, a computer virus here is taken to mean either a virus-A or a virus-B.

The original work with worms done by Shoch and Hupp [14] at Xerox was aimed at harnessing unused resources in computers connected via a network. This work has significant positive implications for parallel computing.

The key difference between a worm and a computer virus is that a worm spreads by replicating itself indefinitely and lives off weaknesses in the host's logic, whereas a computer virus spreads by infecting other programs. The infection occurs each time the Trojan horse instructions are executed, by insinuating itself into the logic of other programs which are then also infected, i.e. the Trojan horse has the write access—although there have been arguments that the write access may not be necessary.

As a worm can replicate itself exponentially, it causes system overloading and degrades system dependability. The consequences of computer virus attack may include files destroyed or contaminated, damage to databases, printing a message on the screen, or a mixture of these. For example, the 'Brain' virus [9,15] has the malicious property of apparently destroying at random part of the File Allocation Table (FAT) which records where files are stored in IBM PC type systems, and changes the volume label on the disk to "(C) Brain". On the other hand, the 'Stoned' virus [15], when executed, displays the message: "Your computer is stoned", and overwrites the FAT on the hard disk.

As a computer virus has the infection property but no ability to replicate itself, the success of a computer virus spread depends on the path of sharing and transitivity of information flow [3,4]. The more significant the sharing of computer systems and the more transitivity there is, the more quickly a computer virus can

spread to other nodes in the network participating in the sharing.

4. Computer virus attacks

As there exist no methods for distinguishing between an unknown computer virus and other programs, i.e. detecting computer viruses *a priori* (this has been proved formally or mathematically [5]), currently the development of a universal computer virus detector (UVD) seems unlikely. In view of this, a cure for all types of computer virus is likely to be extremely difficult. This makes the study of the risk factor of possible computer virus attack more significant.

At present, the existing computer viruses, such as the 'Brain' and 'Stoned', can be regarded as relatively simple types, which mainly strike personal computers and workstations of LANs. However, there is no reason to believe that computer viruses will always be so simple. In the foreseeable future, with the advent of network transmittable C++ programatic objects, and with more and more networks supporting sophisticated services such as remote procedure calls, distributed databases, or network file systems, as well as the common services such as electronic mail, file transfer and remote common execution, computer virus attacks may well become more sophisticated. Widespread attacks by computer viruses on long-haul or wide-area networked systems are already highly possible. Besides, recent advances in very large scale integration (VLSI) have made possible the proliferation of large computer networks in numerous structures across the world. Consequently, containing computer viruses will become a harder task as computer viruses will spread like an epidemic. While in some cases a computer virus may only be extremely inconvenient, in safety-related and reliable computers or in dependable computing with critical applications they could have catastrophic effects. Undoubtedly, the computer virus problem and its threat to computer network dependability is real, and it is said to be growing at an exponential rate.

A computer network is a configuration of two or more computers linked to share information and resources. A computer having the capability to participate in a network is called a node. Computer viruses use the networks as a medium to attack or propagate to local systems. As mentioned before, the attack by computer viruses on WAN has yet to become widespread. Nonetheless, the success of some wide-scale worms (e.g. the 'Christmas Tree' [9] in 1987, and the Internet worm [17] in 1988, both of which hit the headlines of the general press), together with the theory and field experiments of Cohen [3] on computer viruses, clearly indicates that a computer virus attack on WAN, let alone LAN, would be sufficiently simple; although some may argue that the success of the Internet worm was mainly due to the Unix's [1] relatively weak security [7,18]. Incidentally, the Internet worm brought roughly 6000 Unix-based minicomputers and workstations to a halt, costing users an estimated *US*$30 and *US*$92 million in computer time and repairs [2].

5. Risk of a computer virus attack

What is "risk"? In this paper, we regard the risk of a computer virus attack as the probability of such attack resulting in a loss that would accrue to an organization. We postulate that the risk of computer virus attack on a network consists of two elements (c.f. [2]) :

- the probability of completion of computer virus error propagation throughout the network;
- the criticality of the attack, which is defined as the degree of damage the computer virus attack creates on each occurrence.

Here we shall use a mathematical model to predict the probability of completion and the criticality as a result of computer virus error propagation. The costs may, of course, include direct and consequential financial losses, loss of computer and network resources, and compromise of systems performance.

6. Mathematical notation and description

Before proceeding to formulate a model for computer virus error propagation, we must define the notation which will be used in this paper:

- n: number of nodes in the network,

[1] Unix is a trademark of AT&T.

[2] An estimate quoted in the US media.

- ω: mean number of communications between any two nodes in the network per unit time per node,
- $S(t)$: number of susceptible nodes at time t,
- $I(t)$: number of infected nodes at time t,
- $R(t)$: rate at which the susceptible nodes are infected,
- $P_i(t)$: probability that there are i infected nodes at time t,
- $P_n(t)$: probability of completion of computer virus error propagation.

7. Analytical model

As stated previously, direct study of computer virus attacks on real systems is not always feasible since a computer virus can spread quickly, thus possibly causing extensive damage and incurring high cost. Hence, using a mathematical model to study computer virus error propagation is a logical choice.

In this section, a model of the computer virus error propagation process is developed. While building the model, care is taken in view of the fact that considerable variations may occur in computer virus error propagation. A stochastic model describing the change in time of a certain distribution is chosen as it is much more appropriate than a deterministic one in which all mathematical and logical relationships between the elements are fixed. In the model, we assume that computer viruses propagate in a typical network where there is a high degree of homogeneity and that initially one node in the network is infected.

Because the computer virus infection depends on sharing and transitivity as mentioned before, the rate of the computer virus error propagation will depend on the number of communications between any two nodes. Hence, in general, we assume that the rate of computer virus error propagation is proportional to the product of the number of communications between any two nodes in the network and the proportion of the communications which are infective. Other assumptions adopted are :

(1) After an effective communication with an infected node, a susceptible node becomes immediately infective itself, in view of the fact that the Trojan horse is inserted into the programs which are now also infected.

(2) No nodes already infected will be isolated or removed during the period of study; this can be justified by the fact that the finding of a detection mechanism for a computer virus infection (especially an unknown one) within a short time is likely to be difficult, and further that the infection or error propagation rate is probably greater than the detection rate.

(3) There are free communications between any two nodes, i.e. any two nodes in the network are accessible to each other through either a direct link or a path.

(4) No evolution or mutation of a computer virus takes place; this is to make the model less intractable.

From assumption (3) above, we may assume generally that ω is proportional to n, i.e. $\omega = \beta n$ where β is what may be termed the network communication proportionality factor.

Therefore, applying the above assumptions and notation, the stochastic process describing the computer virus error propagation can be represented by the following equations and diagrams. Here, $I(0) = 1$, since initially there is only one infected node; and $R(t)$ = (total number of communications the susceptible nodes have per unit of time) $\times$ (proportions of these communications which are infective). Therefore, we have

$$R(t) = \omega S(t) I(t)/n$$

or equivalently

$$R(t) = \beta S(t) I(t)$$

where β can now be interpreted as an infection parameter.

The rate $R(t)$ reaches the maximum value when $I(t) = n/2$. Thereafter, it decreases, as can be seen in Fig. 1. This is not surprising because the infected nodes keep communicating with nodes which have already been infected.

The state transition diagram for the model of computer virus error propagation through the network is simply shown in Fig. 2 [3].

[3] For notation used in this figure, see Section 6.

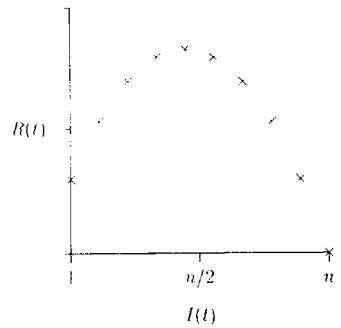

Fig. 1. Rate of infection vs. number of infected nodes at time t for a fixed β.

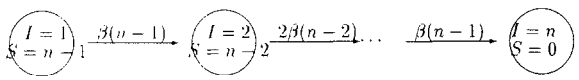

Fig. 2. State transition diagram for the computer virus propagation model.

8. Solution

Based on the model developed above, we shall here look at a solution method, and an example associated with the study of the latency problem posed by computer viruses.

From the previous section, $R(t) = \beta S(t) I(t)$. Thus the probability of transition from any State i to its adjacent State $i+1$ in the state transition diagram given in Fig. 2, during the interval $(t, t+h)$, is given by $\beta S(t) I(t) h + o(h)$, where $(o(h)/h) \longrightarrow 0$ as $h \longrightarrow 0$, i.e. the higher-order terms in h are assumed to be negligible. This leads to the following system of differential equations:

$$\dot{P}_1(t) = -\beta(n-1) P_1(t) \tag{1}$$

and for $2 \le i \le n$,

$$\dot{P}_i(t) = \beta(i-1)(n-i+1) P_{i-1} - \beta i(n-i) P_i(t), \tag{2}$$

and the boundary condition:

$$P_1(0) = 1.$$

By using an iterative method proven in [13], the solution to the above system is found to be (for $i = 1, 2, \ldots, n$)

$$P_i(\tau) = \sum_{j=1}^{n+1} c(i+1, j) e^{b_j \tau} \tag{3}$$

where

$$\tau = \beta t, \text{ [4]}$$

$$b_j = -(n-j+1)(j-1),$$

$$c(i,j) = \begin{cases} 0, & i < j, \\ 0, & i \ge j = 1, \\ 1, & i = j = 2, \\ -\Phi(i), & i = j > 2, \\ \Theta(i,j), & i > j \ne 1, \end{cases} \tag{4}$$

$$\Phi(i) = \sum_{k=1}^{i-1} c_1(i,k),$$

$$\Theta(i,j) = (n-i+2)(i-2)\delta(i-2) \times (c_1(i-1,j)\zeta(b_j - b_i) - c_2(i-1,j)(\zeta(b_j - b_i)^2) + c_2(i-1,j)\zeta(b_j - b_i)\tau),$$

$$\delta(x) = \begin{cases} 1, & x \ge 0, \\ 0, & x < 0, \end{cases}$$

$$\zeta(x) = \begin{cases} \tau, & x = 0, \\ 1/x, & x \ne 0. \end{cases}$$

Note that $c_1(i,j)$ and $c_2(i,j)$ are defined recursively in i, and satisfy the following equation (for fixed j):

$$c(i,j) = c_1(i,j) + c_2(i,j)\tau,$$

with $c_1(i,j)$ being independent of τ, and $c_2(i,j)$ being the coefficient of τ. Therefore, we can use computer implemented algorithms to compute this for various values of i and τ. As the algorithms for computing $c(i,j)$ and $P_i(\tau)$ are straightforward, we shall not include them here.

9. Sensitivity analysis

In order to gain an insight into the nature of computer virus error propagation in a network, the following studies were carried out:

[4] $\tau = \beta t$ is a useful transformation when β is, in practice, likely to be difficult to obtain.

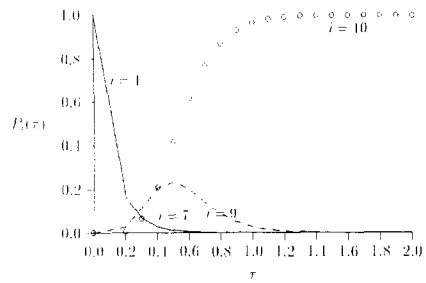

Fig. 3. $P_i(\tau)$ vs. τ for different numbers of infected nodes and $n = 10$.

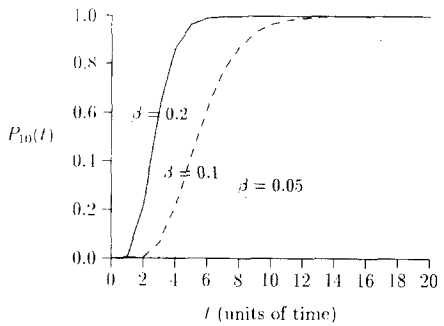

Fig. 4. $P_{10}(t)$ vs. t for different values of β.

- Distribution of the probability of infection with respect to the number of infected nodes in a network;
- Effect of the infection parameter (β) on the probability of a certain number of infected nodes;
- Effect of the number of nodes in the network on the probability of completion of computer virus error propagation.

The above experiments are designed to study the sensitivity of $P_i(\tau)$ or $P_n(\tau)$ to the parameters, such as the number of nodes in the network (n), the infection parameter (β) and the number of infected nodes (i). An understanding of the results of the above studies can help in exploring and planning strategic controls against any computer virus attack on a networked system. The results obtained from these studies are shown in Figs. 3–5.

9.1. Discussion of results

Figure 3 shows that $P_1(\tau)$ decreases at a fast rate and it is most unlikely that there is still only one infected node in the network after $\tau = 0.6$ (for $n = 10$).

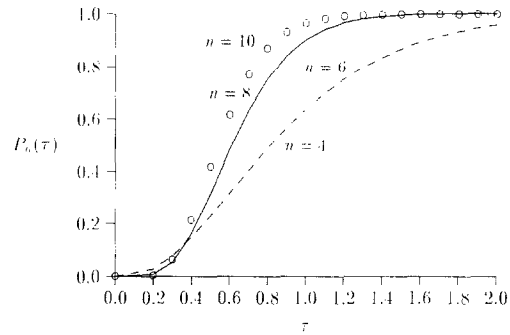

Fig. 5. Probability of completion vs. τ for different numbers of nodes.

While the values of $P_i(\tau)$ for $i = 2, 3, \ldots, 9$ are relatively low throughout, the probability of completion, $P_{10}(\tau)$ increases rapidly until it almost reaches 1 at $\tau = 1.0$. Figure 4 shows that doubling the value of β shortens the time of completion of computer virus propagation by about 8 units of t (for $n = 10$). Figure 5 shows that the rate of change in the probability of completion increases until $P_n(\tau)$ almost reaches 1 as the number of nodes in the network increases.

10. Criticality

We define a measure of criticality, $C(\tau)$ by the following equation:

$$C(\tau) = E(K)P_n(\tau) \tag{5}$$

where K is the total cost resulting from the computer virus attack on the network and $E(K)$ is the expectation of K. Although K includes those cost components (e.g. cost of inconvenience caused by computer virus attack) which are more difficult to quantify, to compute K, we use the following formula:

$$K(T) = \int_0^T f(I(t))\,\mathrm{d}t$$

where $f(\cdot)$ is a known real-valued function defined over the positive integers, and T, a random variable, is the duration of the computer virus propagation completion over the network, i.e. $P_n(\beta T) = 1$. If the costs were to include

(1) Direct and consequential financial losses (including clean-up, repairs, restoration, production disruption and business interruption costs);

(2) Loss of computer and network resources (including number of computer-hours lost),

then we could assume $f(I(t)) = aI(t) + b$, where a and b are given positive constants. Further, we could associate T and another random variable called the stochastic integral defined by

$$H(T) = \int_0^T I(t)\,\mathrm{d}t$$

with their respective costs as follows:

$$\begin{aligned} K(T) &= \int_0^T f(I(t))\,\mathrm{d}t \\ &= \int_0^T (aI(t) + b)\,\mathrm{d}t \\ &= aH(T) + bT. \end{aligned}$$

By using the methods in [11] and [6], the *Laplace Transform* of K is given by

$$\mathcal{K}(s) = \prod_{j=1}^{n-1} \left(1 + \frac{s(a(n-j)+b)}{j(n-j)}\right)^{-1}$$

from which we obtain the following representation for K:

$$K = \sum_{j=1}^{n-1} X_j$$

where each of the random variables X is independent and each has an exponential distribution with parameter $j(n-j)/(a(n-j)+b)$. Hence, the expected value of K is

$$E(K) = \sum_{j=1}^{n-1} \frac{a(n-j)+b}{j(n-j)}$$

and on simplification,

$$E(K) = (a + 2b/n) \sum_{j=1}^{n-1} j^{-1}.$$

Finally, from Eq. (1), the criticality can be expressed as (for $n \geq 2$)

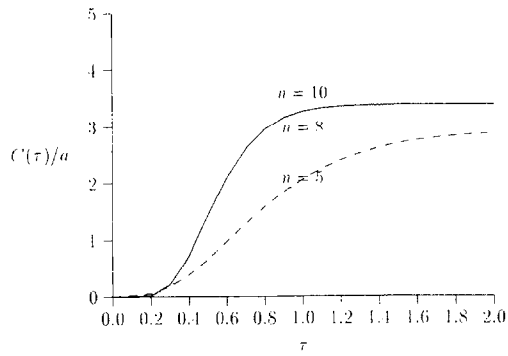

Fig. 6. $C(\tau)/a$ vs. τ for different numbers of nodes and $a : b = 1$.

$$C(\tau) = (a + 2b/n)(1 + 1/2 + 1/3 + \cdots + 1/(n-1))P_n(\tau)$$

or with normalization

$$(C(\tau)/a) = (1 + \frac{2b}{na})(1 + 1/2 + 1/3 + \cdots + 1/(n-1))P_n(\tau).$$

The given values of a and b vary from organization to organization, depending on the magnitude of the direct and consequential losses and the loss of computer and network resources. Intuitively, a and b will also depend on the number of nodes, n, i.e. the greater the value of n, the greater the values for a and b are. As mentioned before, the probability of completion, $P_n(\tau)$, can be obtained from the mathematical model developed in Section 7.

10.1. Studies of normalized criticality

In order to assess the usefulness of the model developed in this paper, sensitivity studies of the normalized criticality $(C(\tau)/a)$ of the following two factors were carried out:

- the number of nodes on the network (n); and
- the cost components ratio $(a : b)$.

The results obtained are shown in Figs. 6–9.

Figures 6–8 show that if the cost component a is greater than the cost component b, then the effect of the number of nodes (n) on the normalized criticality, $C(\tau)/a$, is more significant. On the other hand, Fig. 9 shows that if b is greater than a, then the normalized criticality is almost always greater, as compared to the case when $(a : b) \geq 1$ (for $n = 10$).

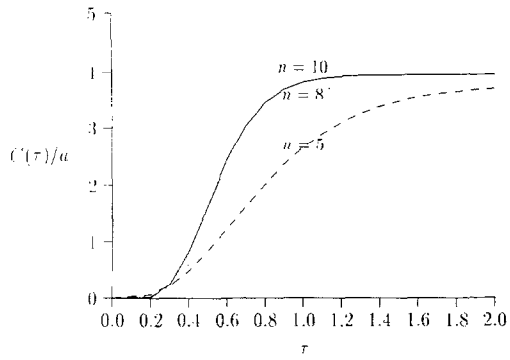

Fig. 7. $C(\tau)/a$ vs. τ for different numbers of nodes and $a : b = 1 : 2$.

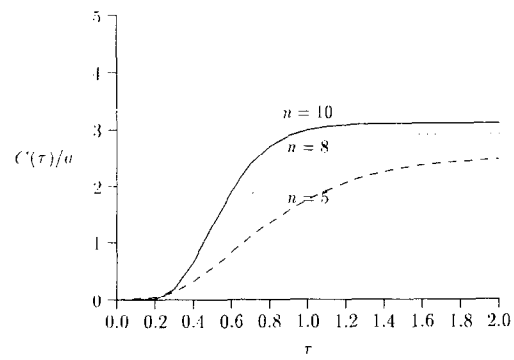

Fig. 8. $C(\tau)/a$ vs. τ for different numbers of nodes and $a : b = 2 : 1$.

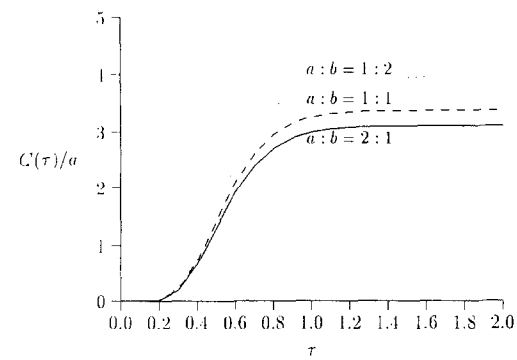

Fig. 9. $C(\tau)/a$ vs. τ for different ratios of $a : b$ and $n = 10$.

Although both a and b are based on static values, Figs. 6–9 reinforce what one would expect in the relative value of network losses when there is a computer virus attack on a network. In particular, Fig. 9 emphasizes network service loss over individual capital losses (such as corruption of a single node or workstation).

11. Strategic controls

Only after having quantified the risk of computer virus attack on a network, can an organization determine what controls are justified, and consider the most effective method of dealing with any computer virus attack. In planning strategic controls against a possible computer virus attack, the organization may consider setting $(C(\tau)/a) = 1$ as the benchmark. The factors a, b and $P_n(\tau)$ can be reduced by controls, but then again the organization must bear in mind the cost of controls. First, prevention measures against computer virus attack cost money. Second, stricter security measures (such as limited sharing, limited transitivity, or even isolating of all the systems) may cause fear that the system will become too unwieldy or difficult for end users, although these measures will of course improve the criticality tremendously. There has obviously to be a balance between risk taken and reasonable control.

However, quantitative specification of values for a and b could be expensive (both in specification and computation) due to real-world variability. Rather, the constants a and b would be multi-valued functions, depending on factors such as the precise loss of network functionality observed (e.g. bandwidth, data, reachability, etc.). Approximations for each individual loss factor (e.g. bandwidth) could be used in the criticality analysis and then simulation experiments could be carried out to select the least costly outcome, or at least to identify behaviors with comparative risk.

12. Conclusion and future work

This paper presented the results of study on the quantitative risk assessment of computer virus attacks on a typical computer network. As a cure for all types of computer virus is likely to be extremely difficult in the near future, the results have much significance for managerial planning by organizations for computer network dependability, particularly with regard to computer safety and network security threatened by

computer viruses, where the cost justification of controls is a major consideration.

The model developed in this paper supports theoretically what is commonly performed in network management, where particular network resources are to be protected or preserved under hostile conditions. Resources are partitioned or separated into multiple areas, so that unwanted behaviors can be damped by boundary (between areas) restrictions. Such behaviors include not only computer virus propagation but other behaviors which have an impact on network stability (e.g. routing and multicasting instabilities). Analysis of preventive measures (such as the use of firewalls) against a possible computer virus attack on a typical network may require the specification of a containment model, where the degree of interconnectivity and thus the value of β are of interest. This is a subject of future research.

References

[1] L.M. Adleman, An abstract theory of computer viruses, in: S. Goldwasser (Ed.), *Advances in Cryptology – Crypto '88. Proceedings*, Lecture Notes in Computer Science 403, Springer, Berlin, 1990, pp. 354–374.

[2] D.M. Chess, Computer viruses and related threats to computer and network integrity, *Comput. Networks & ISDN Syst.*, **17**, (1989) 141–148.

[3] F. Cohen, Computer viruses, Ph.D. Thesis, University of Southern California, 1985.

[4] F. Cohen, Computer viruses: theory and experiments, *Comput. & Security* **6** 22–35, 1987.

[5] W.F. Dowling, There are no safe virus tests, *Am. Math. Monthly* (November 1989) 835–836.

[6] J. Gani and D. Jerwood, The cost of a general stochastic epidemic, *J. Appl. Prob.*, **9** (1972) 257–269.

[7] S. Garfinkel and G. Spafford, *Practical UNIX Security*, O'Reilly & Associates (1992).

[8] K. Gorski et al., Protecting against viruses and designing virus-resistant operating systems, *IFAC Proc. SAFECOMP'90*, October 1990, pp. 43–48.

[9] L.J. Hoffman (Ed.), *Rogue Programs: Viruses, Worms, and Trojan Horses*, Van Nostrand Reinhold, New York (1990).

[10] IEEE Computer Society, *Computer* (July 1989) 84.

[11] D.R. McNeil, Integral functionals of birth and death process and related limiting distributions, *Ann. Math. Statist.*, **41** (1970) 480–485.

[12] M. Pozzo and T. Gray, An approach to containing computer viruses, *Comput. & Security* **6** (1987) 321–331.

[13] N.C. Severo, Two theorems on solutions of differential-difference equations and applications to epidemic theory, *J. Appl. Prob.* **4** (1967) 271–280.

[14] J.F. Shoch and J. Hupp, The worm programs: early experience with a distributed computation, *Comm. ACM* **25** (3) (1982) 172–180.

[15] A. Solomon, *PC Viruses: Detection, Analysis and Cure*, Springer, Berlin (1991).

[16] E. Spafford, The Internet worm program: an analysis, *Comput. Comm. Rev.* **19** (1989) 1.

[17] E. Spafford, The Internet worm incident, Tech. Rep. No. CSD-TR-933, Comp. Sci. Dept., Purdue Univ., 1991.

[18] P.H. Wood and S.G. Kochan, *Unix System Security*, Hayden (1990).

B.C. Soh is a faculty member of Department of Computer Science and Computer Engineering at La Trobe University, and a member of IEEE and its affiliates: Computer Society and Reliability Society. His main research areas include network security, system intrusion detection, malicious software, system dependability evaluation, and fault-tolerant, secure and safe computing.

T.S. Dillon received a Ph.D. degree in Monash University in 1974. He is now the Professor and Chairman of Computer Science and Computer Engineering at La Trobe University and a senior member of IEEE. He is Editor-in-Chief of International Journal of Computer Systems Science and Engineering and International Journal of Engineering Intelligent Systems. He has published more than 400 research papers and four books and made significant contributions to a number of research areas including expert and intelligent systems, neural networks, objected oriented computing, computer communications, and fault-tolerant, secure and safe computing.

Phil County has a B.Sc(Hons) and a Dip. Education, with majors in Computer Science and Mathematics. Following postgraduate research and teaching at the University of Queensland he has been at La Trobe for 11 years where he is the Network Manager and a Lecturer in Computer Science. His research interests are in theoretical and applied network systems engineering, in particular network management models, integrated multimedia network delivery, and Petri Nets.