The Impact of Countermeasure Propagation on the Prevalence of Computer Viruses

Li-Chiou Chen, *Member, IEEE,* and Kathleen M. Carley

***Abstract*—Countermeasures such as software patches or warnings can be effective in helping organizations avert virus infection problems. However, current strategies for disseminating such countermeasures have limited their effectiveness. We propose a new approach, called the Countermeasure Competing (CMC) strategy, and use computer simulation to formally compare its relative effectiveness with three antivirus strategies currently under consideration. CMC is based on the idea that computer viruses and countermeasures spread through two separate but interlinked complex networks—the virus-spreading network and the countermeasure-propagation network, in which a countermeasure acts as a competing species against the computer virus. Our results show that CMC is more effective than other strategies based on the empirical virus data. The proposed CMC reduces the size of virus infection significantly when the countermeasure-propagation network has properties that favor countermeasures over viruses, or when the countermeasure-propagation rate is higher than the virus-spreading rate. In addition, our work reveals that CMC can be flexibly adapted to different uncertainties in the real world, enabling it to be tuned to a greater variety of situations than other strategies.**

***Index Terms*—Computational modeling, computer security, computer virus, netwrok topology, simulation.**

I. Introduction

COMPUTER virus[1] infection problem has imposed significant financial losses as well as the loss of productivity for organizations even though most of these organizations have installed antivirus software. CSI/FBI Survey [8] estimates that the average annual loss due to virus infections is about 283 thousand dollars per organization, 90% of which have installed antivirus software. ICSA Survey [13] reports that virus infections have caused server down time, loss of productivity and loss of data for organizations, 92% of which have installed antivirus software. This evidence implies that installing antivirus software alone cannot resolve the computer virus infection problem effectively unless such software is implemented in the context of a more comprehensive security strategy. If virus countermeasures, such as software patches or new virus definition files, have not been installed on vulnerable computers, these computers can still be infected by new variants of old viruses that exploit the same software vulnerability. How can virus countermeasures be disseminated and installed more effectively than they currently are so that fewer organizations will suffer virus infection problems?

In an attempt to solve this virus infection problem, three antivirus strategies have been proposed. They are 1) the random immunization strategy (RANDOM), 2) the targeted immunization strategy (TARGET) [9], [18], [21], and 3) the kill-signal strategy (KS) [14], [16]. Both RANDOM and TARGET originate from the study of immunization of human populations to prevent epidemics [10]. Neither strategy explains how countermeasures for computer viruses are disseminated. In contrast, KS considers how countermeasures spread but assumes that countermeasures only spread to computers that already have been infected. However, in real-world situations, countermeasures may spread at different rates and through different means of contact than do computer viruses. Further, countermeasures may spread to both infected and uninfected computers thus serving as a preemptive strategy. In order to provide a more effective method for countermeasure propagation in such real situations, we propose an antivirus strategy called the countermeasure competing strategy (CMC).

In this paper, we describe our model for CMC and examine the effectiveness of this strategy by comparing it with the three current antivirus strategies using computer simulation. Note, we conceptualize countermeasures as competing species that act to suppress the spread of computer viruses. Decision makers (representing either people or antivirus software programs) who receive the countermeasures will adopt them (e.g., install new software patches) at a certain rate and spread them with a certain probability.

The rest of the paper is outlined as follows. Section II reviews the theoretical background of our model, gives a brief description of the three current antivirus strategies, and explains our proposed strategy (CMC) in more detail. Section III describes the models for CMC and the simulation tool we have developed. Section IV presents the results from analyzing empirical virus reporting records. Section V describes virtual experiments

Manuscript received March 18, 2002; revised April 7, 2003. This work was supported in part by the NSF/ITR 0218466 and the Pennsylvania Infrastructure Technology Alliance. Additional support was provided by ICES (the Institute for Complex Engineered Systems) and CASOS, the Center for Computational Analysis of Social and Organizational Systems (http://www.casos.ece.cmu.edu) at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the Commonwealth of Pennsylvania, or the U.S. government. This paper was recommended by Editor E. Santos.

L.-C. Chen is with the Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: lichiou@andrew.cmu.edu).

K. M. Carley is with the Institute for Software Research International, Department of Engineering and Public Policy, Carnegie Mellon University Pittsburgh, PA 15213 USA (e-mail: Kathleen.Carley@cmu.edu).

Digital Object Identifier 10.1109/TSMCB.2003.817098

[1]A computer virus is a segment of program code that will copy its code into one or more larger "host" programs when it is activated. A worm is a program that can run independently and travel from machine to machine across network connections [7], [23]. In this paper, the term computer virus will refer to both computer viruses and worms since most malicious programs today can propagate themselves in both ways.

to compare the effectiveness of the four antivirus strategies and discusses the results. Finally, contributions and limitations are discussed.

II. Background

The spread of computer viruses is an example of a nonlinear dynamic system, similar to the spread of epidemics in human populations [14], [21]. The Susceptible-Infected-Removed (SIR) model has been widely used to model the spread of epidemics and to study immunization strategies [1], [2], [10]. The SIR model[2] is a "population-level" description of the epidemic diffusion process that categorizes the entire population into three states: susceptible (S), infected (I), and removed (R). In this model, a portion of the susceptible population is infected at a certain rate through contact with the infected population. At the same time, some of the infected population recover at a certain rate and will not be infected again. The limitation with the SIR model is that it only describes the state changes of the population over time and there are no explicit network assumptions in the SIR model. However, implicitly the SIR model assumes that the population is well-mixed. Namely, everyone is connected to everyone else. This is usually not the case in either human or computer networks. Moreover, previous studies have shown that the spread of epidemics and the spread of computer viruses are dramatically affected by the topology of the underlying networks [1], [15], [17]–[21]. Thus, the SIR assumption that the topology is a fully connected network is likely to result in an overestimation of the rate at which epidemics, or in this case, computer viruses, spread. To counter this overestimation, the SIR model requires increasing the number of model variables to account for variations in network structure.

Three antivirus strategies that add network consideration to the SIR model have been proposed: 1) the random immunization strategy (RANDOM), 2) the targeted immunization strategy (TARGET) [21], and 3) the kill-signal strategy (KS)[3] [14]. RANDOM proposes to immunize a certain portion of randomly selected nodes so that the virus will not prevail because the immunized nodes cannot be used to spread viruses. TARGET proposes a similar strategy but immunizes nodes that have high connectivity. Both strategies have been studied for controlling the spread of epidemics in human populations [1] and for controlling the spread of computer viruses through complex networks [9], [18], [21], [24]. KS proposes that once a virus infection is found in a computer, the computer will disseminate countermeasures to other infected computers.

In contrast to these three strategies, we propose the countermeasure competing strategy (CMC). The CMC is based on the hypothesis that the countermeasure for a new computer virus can be spread through a countermeasure-propagation network. The spread of countermeasures is similar to the spread of computer viruses; but, unlike computer viruses that propagate themselves, countermeasures act to suppress the spread of computer viruses. This can be thought of as having two viruses spreading at the same time: a "good" virus and a "bad" virus. Factors that influence the spread of the good one over the bad one enable the overall system to become less vulnerable to the bad virus. The types of countermeasures depend on how CMC is implemented. A common example is a warning disseminated via e-mails that ask people to be aware of new computer viruses or new software vulnerabilities. Another example is to create an automatic mechanism for spreading new software patches. Users who like to adopt the automatic mechanism can install a software program on their computers to authenticate and install the software patches. A similar mechanism has been implemented in most current antivirus software products,[4] but these products only allow a server to disseminate countermeasures to its client computers, which at this point are not able to further distribute the countermeasures. We will not further discuss the specific implementation details of countermeasures since they are beyond the scope of this paper.

All four strategies hold several of the same assumptions about disseminating viruses, which are derived from the SIR model. These are 1) nodes disseminate viruses to neighboring nodes in the network once they are infected and stop once the infection is discovered and 2) viruses will not infect nodes that have adopted countermeasures.

What distinguishes CMC from the other three strategies is its method of disseminating countermeasures. Both RANDOM and TARGET assume that a certain portion of preselected nodes (immunized nodes) adopt countermeasures, but these nodes do not further spread countermeasures. In contrast, both KS and CMC assume that only a very small portion of nodes are preselected and these nodes are able to further spread countermeasures. There are three different assumptions about how countermeasures are disseminating between KS and CMC. First, KS assumes that the adoption of countermeasures is mandatory. CMC assumes that the adoption of countermeasures is probabilistic (modeled by a rate of adoption) which provides a more accurate model of how countermeasures spread in the real world. For example, it is uncertain whether users would actually adopt countermeasures after receiving information about them. Secondly, KS assumes that countermeasures only spread to the nodes that have been infected, but CMC assumes that countermeasures may spread to both susceptible nodes and infected nodes. Again, this assumption allows CMC to set the model closer to real-world situations. The rational for KS excluding susceptible nodes from adopting countermeasures is based on the idea that only infected nodes have a reason to search for adopting countermeasures. In reality, some users are able to and desire to preemptively adopt countermeasures. For example, users may warn friends and associates about an e-mail virus even though they themselves have not encountered it. Those friends and associates may then take preemptive action. Finally, KS assumes that countermeasures and viruses spread through the same network—the network linking computers (and so their users). In contrast, CMC assumes that coun-

[2]In this model, α denotes the infection rate of susceptible population and γ denotes the recovery rate of infected population. Changes of populations in the three states over time can be represented mathematically as $((dS)/(dt)) = -\alpha SI$, $((dI)/(dt)) = \alpha SI - \gamma I$, $((dR)/(dt)) = \gamma I$.

[3]Here we use the same terminology "RANDOM," "TARGET," and "KS" as they are used in [19]–[21].

[4]For example, Symantec, McAfee and Sophos all have products to support this functionality.

TABLE I
NOTATIONS OF MODEL PARAMETERS

Input parameters		
notations	meaning	range of parameter value
G_v	virus-spreading network	undirected graph
G_c	countermeasure-propagation network	undirected graph
α	virus birth rate	[0,1]
γ	virus death rate	[0,1]
ρ_v	virus-spreading rate	$=\alpha/\gamma$
λ	countermeasure birth rate	[0,1]
δ	countermeasure death rate	[0,1]
ρ_c	countermeasure-propagation rate	$=\lambda/\delta$
κ	countermeasure adoption rate	[0,1]
n	percentage of immunized nodes	0-100%
Output parameters		
notations	meaning	range of parameter value
S	size of the virus infection	[0,1]
D	converge time	>0

termeasures spread through a distinct network—the network linking users (and so their computers); whereas, the viruses spread through the network linking computers (and so their users).[5] Note, we are assuming a one-to-one mapping between users and computers, but two types of relations—the computer network and the social network (which may use means other than the computer backbone). In the real world, users often learn about a new virus and its countermeasure through friends, the news media, or their system administrators, not just from the virus victims. In many cases, victims' systems are down and are not capable of warning others of the virus. Although this separate network assumption increases the complexity of the model, it is more realistic for modeling the countermeasure dissemination. In the next section, we will explain our model for CMC.

III. MODELING THE DYNAMICS OF COMPUTER VIRUS PROPAGATION

This section describes the model for CMC and the simulation tool for comparing CMC with the three antivirus strategies described above. Table I lists the notation and meaning of parameters used in the model. Our goal is to model the computer virus infection problem in the real world at an abstract level that can generate useful policy conclusions. We are not trying to create an exact model of the real world.

A. Model for CMC

We define the virus-spreading network G_v as the network for spreading viruses, and the countermeasure-propagation network G_c as the network for spreading countermeasures. Both G_v and G_c are undirected graphs. In the real world, both networks can represent either physical networks (connecting computers/programs) or social networks (connecting people/groups). The real world representation of G_v depends on the vulnerabilities that the virus exploits. For example, if a computer virus, such as Love Letter,[6] spreads through e-mails or mailing lists, G_v is a social network because the virus exploits the social/organizational connections among people/groups that are built upon e-mail communications, in which the virus spreads from one e-mail account (representing one person or one group) to another e-mail account. In contrast, a computer virus such as Nimda exploits specific software vulnerabilities in order to propagate itself without user intervention. In this case, G_v is a physical network connected by vulnerable computers/programs. Similarly, the real world representation of G_c depends on the implementation of antivirus policies. On the one hand, G_c is a social network if countermeasures are implemented as e-mail warnings. On the other hand, G_c is a physical network if countermeasures automatically spread through antivirus programs that have been installed by system administrators on computers beforehand. In summary, the differences between these two complex networks in the model are not exactly the differences between a physical network and a social network. Whether G_v or G_c is a social network or a physical network depends on the vulnerability/information that the virus/countermeasure utilizes in order to spread.

In this paper, we make two simplifying assumptions that facilitate evaluating the effectiveness of CMC. These assumptions can be relaxed in future applications of our model. First, we assume that countermeasures have only a positive effect and no negative effects on the action of decision makers. For example, a software patch is authentic and is not another computer virus. This assumption should be investigated in a future evaluation of CMC because countermeasure propagation can become a computer virus problem if the countermeasures have negative effects. Secondly, we assume that each node in G_v maps to one node in G_c. That is, the two-network model can be though as a network where the nodes are connected by different types of links (that is, a multiplex network).

The changes in each node of these two complex networks are described as state machines. Each node in G_v changes over time among three states: "susceptible (S)," "infected (I)," and "removed (R)" due to the spread of computer viruses, as illustrated in the state machine in Fig. 1. In the meantime, each node in G_c changes among three states: "unwarned (U)," "warning (WG)," and "warned (WD)" due to countermeasure propagation, as illustrated in the state machine in Fig. 2. Since we assume that each node in G_v maps to one node in G_c, we can think of each node as one system being described by two variables: its computer virus state and its countermeasure state. That is, one node can be in either one of the nine cases (3*3, one of three states in G_v* one of three states in G_c). For example, a node can be both "susceptible" and "unwarned." The transition from one case to another is described in the following sub-sections as state transition rules. The two state machines interact with each other

[5]Note, it is functionally equivalent in this case to treat these as two networks versus a single multiplex network (one network with two types of ties). We refer to it here as two network simply for the sake of clarity of exposition.

[6]This virus propagates itself to other e-mail accounts only when e-mail recipients click on the malicious e-mail attachments.

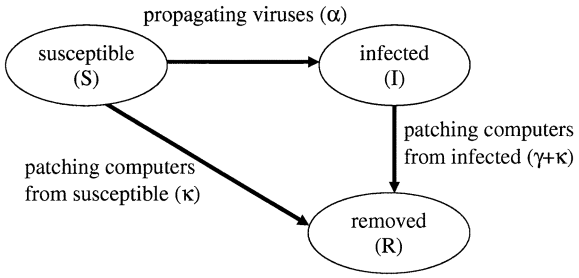

Fig. 1. State machine for computer-virus spreading.

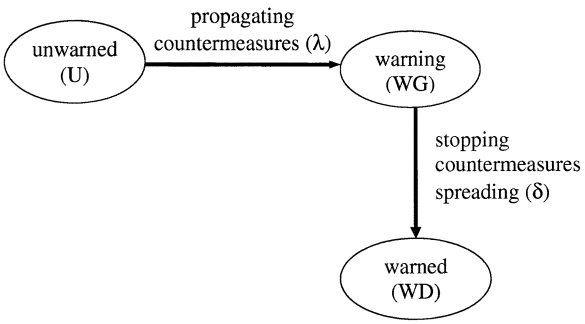

Fig. 2. State machine for countermeasure propagation.

through these rules. Here we describe the two state machines separately for clarity.

In each state machine, circles represent states and arrows represent state transition rules. We label each arrow with the name of the state transition rule and a Greek letter representing the probability of change for each node from one state to another state. All probabilities in state transition rules are in range [0,1]. We describe these states and rules in details as follows.

1) State Machine for Computer-Virus Spreading: As in Fig. 1, each state in the state machine for computer-virus spreading represents an observable fact of a node. A state is a Boolean variable of value either "true" or "false." The state machine is revised from the SIR model, which includes three states:

1) **Susceptible (S):** A node has the software vulnerability that the computer virus can exploit.
2) **Infected (I):** A node is infected by the computer virus, which means the node can infect its neighbors with this virus, and the virus has not been removed from the node. For example, computers that receive a Melissa Virus are in the "infected" state only if the users click on the e-mail attachments and only if the computers can spread this virus.
3) **Removed (R):** A node that has installed a detection tool that identifies and removes a computer virus, or a node that has installed a software patch to eliminate the software vulnerability exploited by a virus.

There are three state transition rules for spreading viruses:

1) **Propagating viruses:** A node in the "susceptible" state will change to the "infected" state with the probability α only if one of its neighbors is infected, where α is the birth rate[7] of the computer virus. Since the decision makers for this node have not adopted countermeasures (the node is susceptible), the state of the node for countermeasure propagation does not matter in this case.
2) **Patching computers from susceptible:** A node in the "susceptible" state will change to the "recovered" state at the probability κ if the corresponding node in G_c is in either the "warning" state or the "warned" state. κ denotes the countermeasure adoption rate and represents the probability that decision makers will adopt the countermeasure.
3) **Patching computers from infected:** A node in the "infected" state will change to the "removed" state at the probability $\gamma + \kappa$ if the corresponding node in G_c is in either the "warning" state or the "warned" state. The "infected" node will change to the "removed" state at the probability γ otherwise. The death rate of the computer virus γ represents the probability that decision makers discover the virus infections and patch the computers. Comparing to the previous rule, this rule implicitly assumes that the "infected" nodes are more likely to adopt countermeasures than the "susceptible" nodes $(\gamma + \kappa > \kappa)$. In order to discuss how fast a virus can spread preemptively when no countermeasure is applied, we define the virus-spreading rate $\rho_{\mathrm{v}} = (\alpha/\gamma)$.

2) The State Machine for Countermeasure Propagation: As in Fig. 2, each state in the state machine of countermeasure propagation represents whether or not a decision maker has adopted and spread countermeasures. This state machine includes three states:

1) **Unwarned (U):** The node has not received countermeasures and will not be influenced by countermeasures.
2) **Warning (WG):** The node has received countermeasures and would further spread the countermeasure at a certain probability.
3) **Warned (WD):** The node has received countermeasures but does not further spread countermeasures anymore.

There are two state transition rules for spreading countermeasures:

1) **Propagating countermeasures:** A node in the "unwarned" state will change to the "warning" state with the probability λ if one of its neighbors is in the "warned" state, where λ is the birth rate of the countermeasure.
2) **Stopping countermeasures spreading:** A node in the "warning" state will change to "warned" state at the probability δ, where δ is the death rate of the countermeasure. We assume that a node stops spreading the countermeasure with a certain probability for two reasons. First, if the countermeasure represents an e-mail warning, people who have received the e-mails may not keep propagating the e-mails all the time. Secondly, if the countermeasure represents a software patch sent by an automatic mechanism, the death rate will prevent the patch spreading from saturating the computer network. This setting eliminates a possible negative effect caused by countermeasure propagation. In order to discuss how fast a countermeasure can spread, we define the countermeasure-propagation rate $\rho_{\mathrm{c}} = (\lambda/\delta)$.

[7]The terms "birth rate" and "death rate" of computer viruses have first been used in [14]–[16].

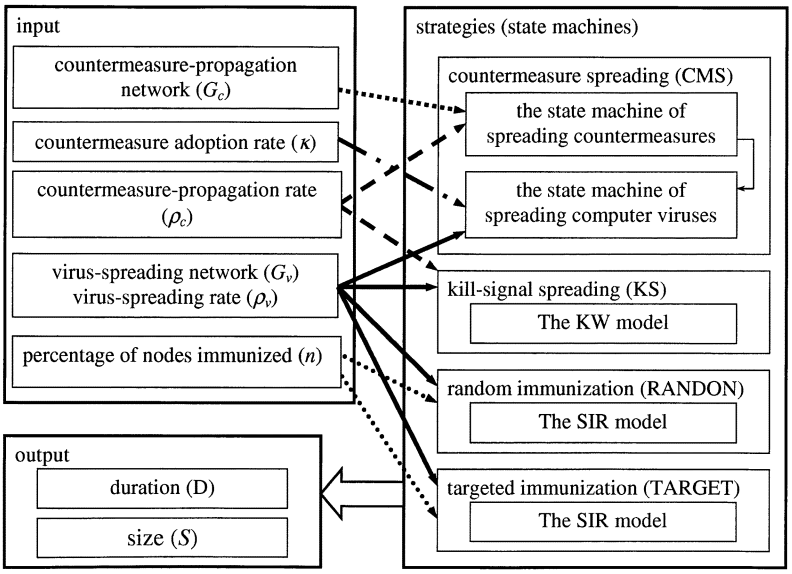

Fig. 3. Simulation of antivirus strategies.

B. Simulation of Antivirus Strategies

The simulation is designed to be flexible enough so that it can examine the effectiveness of the four different antivirus strategies on various network topologies using Monte-Carlo sampling techniques. The four strategies include RANDOM, TARGET, KS, and CMC, as described in Section II. Fig. 3 illustrates the input parameters, the state machines, and the output parameters for each one of the four antivirus strategies.[8]

Using this simulation tool, we conduct several sets of virtual experiments, as described in Section V. Each virtual experiment stops when the dynamic system converges to a steady state. In our simulation, the steady state means that no nodes are in the "infected" state or all nodes have been infected. The outputs include the duration of the virus infection (D) and the size of the virus infection (S).[9] D refers to the time that the system converges. S refers to the number of nodes that have been infected divided by the total number of nodes in the network.

IV. Analysis of Virus Reporting Records

In this section, we calibrate ρ_{v} and G_v based on empirical virus reporting records using The Wild List[10] (TWL). The data set we analyze is from January 1996 to September 2002, which includes 106 reporting sites and 958 computer viruses across 71 reporting[11] time periods. TWL was originally published in one-month chucks. It reports the name of the viruses that have been discovered in each reporting site (a site refers to a company or a reporting center) over time but does not report the number of infected computers in each site. For this reason, this data set is only enough to investigate the prevalence of computer viruses among organizations but not within an organization.

We first estimate ρ_{v} using the TWL data set. We calculate the size of the virus infection (S) and the duration of the virus infection (D) for each virus in TWL data set, and then calibrate ρ_{v} for CMC to match S and D using the simulation tool described in Section III-B. Based on the data, $0.02 \leq \mathrm{S} \leq 0.6$ $(\mathrm{mean} = 0.09)$ and D is around 90 days (three months) because viruses have infected an average of 80% of the sites when the system converges in the first three months of the duration of the virus infection. In this range, the calibrated ρ_{v} is between 0.01 and 0.2 $(\mathrm{mean} = 0.05)$. Since both S and D do not vary significantly with the way that the viruses propagate, we decide not to further categorize viruses and to use the maximum calibrated $\rho_{\mathrm{v}} = 0.2$ ($\rho_{\mathrm{v}} = \alpha/\gamma$ where $\alpha = 0.02$ and $\gamma = 0.1$) as a base case for later virtual experiments. Appendix A is a detailed description of the analysis. The ρ_v calibrated from this data set may be underestimated for two reasons. First, the TWL data set is an observed virus prevalence record in which it is possible that some virus infection incidents are not reported because they have not been discovered. Secondly, the observed prevalence records may be a result of applying some antivirus strategies. For this reason, we examine the variation of ρ_v in the virtual experiments in Section VI.

We then infer the topology of the virus-spreading network from the TWL data set. In order to infer G_v, we assume that two reporting sites have a link to each other if one site reports a virus during the current time period and the other site reports the same virus the first time during the next time period. We code the reporting records for each virus as a network and then obtain a conjunction network from these networks. We call this conjunction network the TWL network (labeled as TWL). Appendix B describes the details of inferring the TWL network from the data set. In the later virtual experiments, the virus-spreading network

[8]We refer to the model used by KS as the KW model [14], which adds an additional state to the SIR model to represent the spread of kill signals (such as virus warnings).

[9]The terms here are borrowed from "duration of the epidemic" and "size of the epidemic", which are commonly used in epidemiological literature. "Duration of the epidemic" refers to the time between the epidemic starts and it converges to a steady state and "size of the epidemic" refers to the fraction of individuals who have been infected with the disease over time.

[10]The Wild List is available at www.wildlist.org. It is also accessible from the Virus Bulletin (www.virusbtn.com/resources/wildlists/index.xml). The Wild List is a cooperative listing of viruses reported as being in the wild by virus information professionals. ICSA, Virus Bulletin and Secure Computing are currently using The Wild List as the basis for virus testing and certification of antivirus products [12].

[11]From 1996 to 1998, The Wild List reported the records every two months.

TABLE II
EXPERIMENT DESIGNS

parameters	experiment 1	experiment 2	experiment 3	experiment 4
virus spreading network (G_v)	TWL	TWL	TWL, SF, LAT, RG, FULL, AS	AS (for SF-L, LAT-L, RG-L, FULL-L) and TWL (for SF, LAT, RG, FULL, SF-S, LAT-S, RG-S)
effective virus spreading rate (ρ_v)	0.01, 0.05, 0.1, 0.2, 0.3 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1	0.2	0.2	0.2
countermeasure propagation network (G_c)	TWL	TWL	TWL, SF, LAT, RG, FULL, AS	SF, LAT, RG, FULL SF-S, LAT-S, RG-S SF-L, LAT-L, RG-L, FULL-L
effective countermeasure propagation rate (in ρ_c/ρ_v)	10 (applicable to CMC and KS)	0.05, 0.5, 1, 2, 5, 10, 20, 50 (applicable to CMC and KS)	10 (applicable to CMC and KS)	0, 0.5, 1, 2, 4, 6, 12
countermeasure adoption rate (κ)	0.1(applicable to CMC)	0, 0.01, 0.05, 0.1, 0.5, 1 (applicable to CMC)	0.1, 0.5 for (CMC)	0.1
percentage of nodes immunized (n)	50% (applicable to RANDOM and TARGET)	1%, 5%, 10%, 20%, 30%, 50%, 70%, 90% (applicable to RANDOM and TARGET)	50% (applicable to RANDOM and TARGET).	N.A.
anti-virus strategy	CMC, KS, RANDOM, TARGET	CMC, KS, RANDOM, TARGET	CMC, KS, RANDOM, TARGET	CMC

G_v is set to the TWL network as a base scenario. G_v calculated from this method represents the worst possible case of computer virus spreading which we will use to examine the lower bound of the four antivirus strategies.

V. VIRTUAL EXPERIMENTS

A. Experiment Design

We design four sets of virtual experiments to compare CMC with other three strategies. Table II lists the values of parameters in each set of experiments. For each experiment, unless it is specified, we use $\rho_v = 0.2$ and $G_v = G_c = \text{TWL}$ network as a base scenario, which is estimated from the TWL data. Each experiment is run 10^5 iterations so that the standard deviations and the mean values of outputs converge. One starting infected node is randomly selected each run.

Experiment 1 compares the four strategies by varying ρ_v so that we can investigate how these strategies perform when viruses spread at various rates. Experiment 2 varies ρ_c and κ in CMC and KS, and varies n in RANDOM and TARGET. The purpose is to realize the constraints for using these strategies. Experiment 3 compares the four strategies using six different networks (for both G_v and G_c) so that we can understand how these strategies differ under various network topologies. The six networks include two empirical networks and four theoretical networks. The two empirical ones are the TWL network (TWL), which has 106 nodes and is inferred from the virus data, and an Internet autonomous system network topology[12] (AS), which has 11 716 nodes and is used to examine how the number of nodes modeled on the effectiveness of the four strategies. The four theoretical ones include a scale-free network (SF),[13] a lattice (LAT)[14], a random network (RG), and a fully connected network (FULL). The four theoretical ones are calibrated to be the same size and have approximately the same number of links as TWL.[15] In experiment 4, we concentrate on investigating what properties of the countermeasure-propagation network influence the effectiveness of CMC. In this experiment, we fix G_v to one of the empirical networks and vary G_c to one of the four theoretical networks. All theoretical networks have the same number of nodes as the corresponding empirical network but are designed to have varying network properties for comparison purposes. Properties of networks used in experiment 3 and 4 are listed in Table III.

We will discuss the results from experiment 1 and 2 in Section V-B and the results from experiment 3 and 4 in Section V-C. Results are reported in either S or RS. S is calculated as the average value of 10^5 runs. RS is calculated as the ratio of S based on one strategy to S without any antivirus strategy. For clarity of presentation, the antivirus strategies are labeled with their model parameters in the next two sub-sections. They are $\text{CMC}(\rho_c/\rho_v, \kappa)$[16], $\text{KS}(\rho_c/\rho_v)$, $\text{RANDOM}(n)$, and $\text{TARGET}(n)$. A model parameter is labeled with an asterisk to represent an independent variable in the analysis. "No strategy" means no antivirus strategy is applied.

B. Result Discussions: The Impact of Model Parameters

First, we are interested in comparing CMC to the three antivirus strategies under various virus-spreading rates. Fig. 4 (from experiment 1) shows how much each antivirus strategy reduces the size of virus infection when virus-spreading rate is varied. When ρ_v is in the range estimated from the TWL data (between 0.01 and 0.2), the order of reducing S from the most to the least is $\text{TARGET}(50\%) > \text{RANDOM}(50\%) > \text{CMC}(10, 0.5) > \text{CMC}(10, 0.1) > \text{KS}$. Once ρ_v increases past 0.15, CMC(10,0.5) reduces S more than RANDOM(50%).

Although TARGET(50%) reduces S the most across ρ_v, immunizing 50% of nodes in the population could be costly in the real world. Fig. 5 (from experiment 2) shows how S varies in both TARGET and RANDOM when n varies. Comparing Fig. 5 with Fig. 4, we find that CMC(10,0.1) reduces S more than TARGET and RANDOM when only 20% of nodes are immunized. To further investigate CMC with different parameter settings, Fig. 6 illustrates how S varies in both KS and CMC when ρ_c varies. Comparing Fig. 6 with Fig. 4, we find that CMC reduces S more than RANDOM(50%) only when $\kappa > 0.1$ and $\rho_c/\rho_v \geq 10$.

In summary, each of the four strategies has its strength and weakness. It is important to realize the constraints before applying them to the computer virus infection problem. Both RANDOM and TARGET require a few nodes to be immunized

[12]Available at "http://moat.nlanr.net/AS/", downloaded on August 2001.

[13]All scale-free networks are generated based on the algorithm in [4].

[14]We use the Small-World network algorithm in [26] to generate the lattice (with $\text{reconnecting probability} = 0$), and the random network (with the $\text{reconnecting probability} = 1$).

[15]The number of links cannot be exactly the same due to the topology of the network. In particular, the fully connected network cannot have the number of links.

[16]For example, CMC(10,0.1) represents CMC has countermeasure-propagation rate 10 times of virus-spreading rate, and countermeasure adoption rate 0.1.

TABLE III
PROPERTIES OF NETWORKS USED IN SIMULATION EXPERIMENTS

sources of networks	label	number of nodes	number of edges	density	average path length	clustering coefficient	degree centralization	epidemic threshold
empirical	TWL	106	2710	0.24	1.5	0.77	3.78E-03	0.016
	AS	11716	24480	1.8E-04	3.6	0.30	1.80E-05	3.7E-03
theoretical	LAT	106	2650	0.24	1.6	0.73	0.00E+00	0.020
(experi-	RG	106	2650	0.24	1.5	0.47	8.32E-04	0.020
ment 3)	SF	106	2745	0.25	1.5	0.67	2.85E-03	0.017
	FULL	106	11130	1.00	1.0	1.00	0.00E+00	0.01
theoretical	LAT-S	106	212	0.02	13.6	0.50	0.00E+00	0.25
(experi-	RG-S	106	212	0.02	3.4	0.04	6.47E-04	0.21
ment 4)								
(for TWL)	SF-S	106	208	0.02	3.1	0.10	1.58E-03	0.15
theoretical	LAT-L	11716	23432	1.7E-04	1464.9	0.50	0.00E+00	0.25
(experi-	RG-L	11716	23432	1.7E-04	7.3	1.5E-04	0.00E+00	0.22
ment 4)	SF-L	11716	23428	1.7E-04	5.1	4.9E-03	2.00E-06	0.07
(for AS)	FULL-L	11716	1.4E+08	1.00	1.0	1.00	0.00E+00	8.5E-05

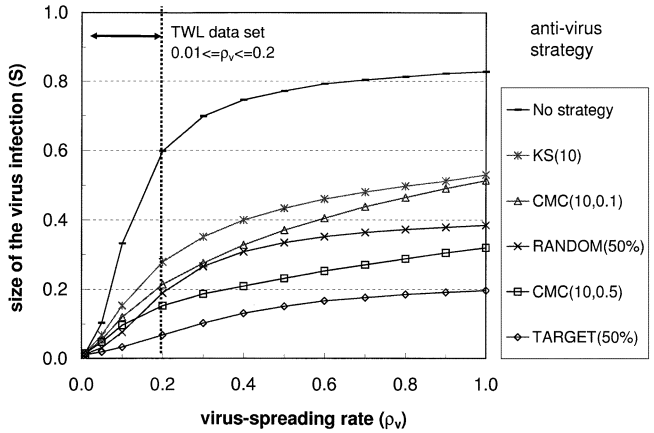

Fig. 4. Effectiveness of CMC comparing with KS, RANDOM, and TARGET when ρ_v varies.

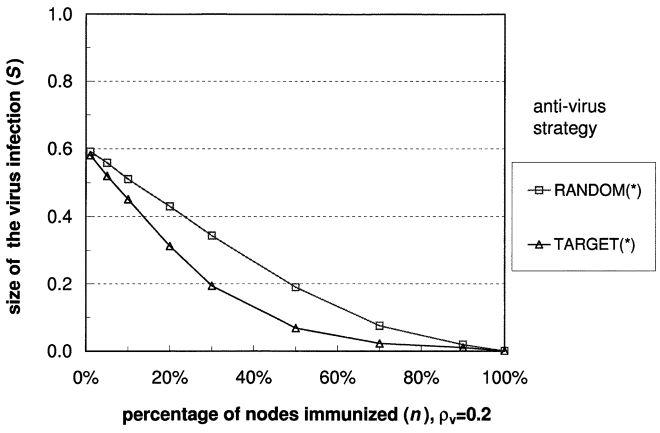

Fig. 5. Effectiveness of RANDOM compares with TARGET when n varies.

before a virus infects them. In the real world, immunizing a large portion of nodes could be costly. Although TARGET can reduce the size of the virus infection to the same level as RANDOM does by immunizing fewer nodes, as shown in Fig. 5, it is hard to determine which nodes have high connectivity because computer viruses operate through many different networks where one node may be highly connected in a network but not in another network. In this aspect, RANDOM is more applicable to the real world than TARGET.

In contrast to RANDOM and TARGET, both KS and CMC focus on distributing countermeasures for a virus without immunizing a large portion nodes beforehand. The idea of propagating countermeasures through a network gives both KS and CMC an advantage over TARGET where one must identify highly connected nodes before the attack. At the same ρ_C, CMC reduces S more than KS does because of the different

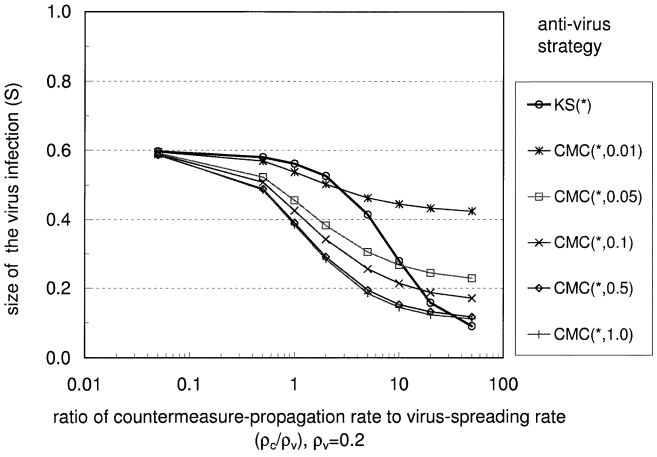

Fig. 6. Effectiveness of CMC compares to KS when ρ_v and κ vary.

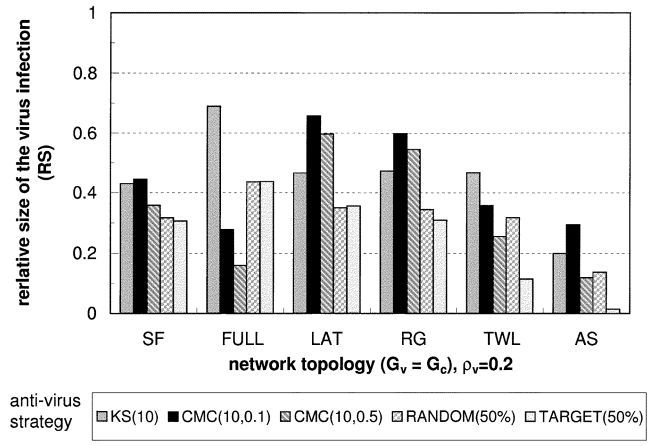

Fig. 7. Effectiveness of CMC comparing with KS, RANDOM and TARGET when the network topology varies ($G_v = G_c$).

assumptions described in Section II. Based on these assumptions, CMC incorporates uncertainty in how susceptible nodes are immunized and how countermeasures are spread into the strategic response to virus attacks. In addition, CMC identifies two limitations of using the spread of countermeasures to suppress the spread of computer viruses. One limitation is that countermeasures have to spread faster than computer viruses ($\rho_v/\rho_c > 1$) and another limitation is that the probability of adopting countermeasures at each node must be greater than 0.1 ($\kappa > 0.1$, explicitly, KS assumes $\kappa = 1$).

C. Result Discussions: The Impact of Network Topology

The topology of G_v may vary from one virus to another. For example, G_v for a virus spreading through e-mails is different from G_v for a virus spreading through web browsing. Similarly, the topology of G_c may vary from one antivirus policy to another. For example, G_c for sending e-mail warnings is different from G_c for sending software patches by system administrators. The variation of networks in the real world is the reason why we need to study the impact of network topology on the effectiveness of the four antivirus strategies.

First, we ask how the network topology influences the effectiveness of CMC comparing to the other strategies. Fig. 7 (from experiment 3) compares CMC(10,0.1) and CMC(10,0.5) with KS(10), RANDOM(50%), and TARGET(50%) for six different network topologies. CMC(10,0.5) reduces RS the most for FULL and the second for TWL and AS. CMC performs the worst under LAT and RG topologies among the strategies. The results imply that the effectiveness of CMC may be dependent on some properties of the networks. To further confirm this observation, we run experiment 4 to correlate the effectiveness of CMC to a set of network properties.

As in experiment 4, we vary G_c. in order to investigate what properties in a countermeasure-propagation network actually influence the effectiveness of CMC. As in Table IV, we correlate the properties of networks to RS and find that the correlation varies with both ρ_c and the properties of networks.

Among the properties we calculated, epidemic threshold has the highest positive correlation to RS when ρ_c is larger than $\rho_v(\rho_c/\rho_v > 1)$. Epidemic threshold is defined as the minimal epidemic spreading rate that an epidemic can prevail [1]. In a complex network, epidemic threshold varies with the edge distribution of networks[17] [19]. Applying this property on countermeasure propagation, we find that the countermeasure-propagation network with a lower epidemic threshold is more effective in reducing the size of the virus infection than networks with higher epidemic thresholds.

TABLE IV
CORRELATIONS BETWEEN PROPERTIES OF COUNTERMEASURE-PROPAGATION NETWORKS (G_c) AND RELATIVE SIZE OF THE VIRUS INFECTION (RS)

	The ratio of countermeasure-propagation rate to virus spreading rate (ρ_c/ρ_v)						
	0	0.5	1	2	4	6	12
epidemic threshold	0	0.65	0.84	0.93	0.94	0.92	0.92
density	0	-0.98	-0.86	-0.71	-0.58	-0.51	-0.49
average path length	0	0.24	0.30	0.36	0.47	0.56	0.64
clustering coefficient	0	-0.83	-0.82	-0.68	-0.52	-0.42	-0.36
degree centralization	0	-0.75	-0.55	-0.25	-0.18	-0.18	-0.22

In addition, density[18] has a negative correlation with RS. This result implies that our strategy is more effective if the connectivity of G_c is larger. Moreover, the effectiveness of CMC increases with clustering coefficient[19] (negatively correlated to RS), and decreases with average path length (positively correlated to RS). This result confirms the finding in [26] about epidemic spreading across a network with the Small-World property. Finally, we found that the effectiveness of CMC increases when the degree centralization[20] [25] of a network increases. However, the correlation is smaller comparing to other properties.

In summary, we find that CMC is more effective than the other three antivirus strategies when the countermeasure-propagation networks are highly connected (as such FULL) or highly centralized (with a lower epidemic threshold, a higher clustering coefficient, or a shorter average path length). If the topology of G_v can be determined, we can design G_c as the network that can spread countermeasures faster than the computer viruses, such as a network with a lower epidemic threshold. If the topology of G_v cannot be determined, increasing ρ_c or κ is another way to reduce the size of the virus infection because the effectiveness of CMC increases when either of the two variables increases.

[17]When an epidemic spreads on a complex network, the epidemic threshold can be estimated by $\rho_{threshold} = (\langle e\rangle/\langle e^2\rangle)$ where $\langle e\rangle$ denotes the average number of edges and $\langle e^2\rangle$ denotes the average square of edges [19].

[18]Density measures the connectivity of a network, which is defined as the number of edges of a network divided by the largest possible number of edges of this network [25].

[19]Clustering coefficient measures the cliquishness of a network. Node clustering coefficient is defined as the connectivity of the neighbors of a node. Clustering coefficient is the average of node clustering coefficients in a network [26].

[20]Degree centralization measures the differences of the connectivity among nodes, which takes the average of the difference of individual node connectivity and the average node connectivity [25]. Degree centralization can be used as an index only if it is larger than 1 because all graphs that have the same number of edges per node have $\mathrm{degree\ centralization} = 1$. For example, both a fully connected network and a lattice network have $\mathrm{degree\ centralization} = 1$. For this reason, this index cannot distinguish edge distribution among nodes well when it is equal to 1.

VI. Conclusions

In this paper, we propose a new antivirus strategy—the countermeasure competing strategy (CMC)—for mitigating the severity of impact of computer-virus infections. We investigate the effectiveness of this strategy by comparing it, via computer simulation, with three antivirus strategies previously discussed in the literature—RANDOM, TARGET and KS. Our results demonstrate that CMC is the most effective strategy in general and when the networks are constrained to match the empirical data on virus spreading. CMC is based on the idea that countermeasures against computer viruses can spread as competing species on a separate network from the network used to spread computer viruses. By using CMC, we find the size of the virus infection can be reduced significantly only when countermeasures spread faster than computer viruses.

How can virus countermeasures be disseminated and installed more efficiently than they currently are so that fewer organizations will suffer virus infection problems? We believe our analysis provides several insights into this problem. First, one of the reasons that CMC is more effective than KS is that CMC spreads countermeasures to both susceptible nodes and infected nodes. In real-world responses to outbreaks of computer viruses, antivirus companies or computer incident response teams should spread warnings (behavior countermeasures) or software patches to their customers as soon as possible whether or not they have been infected. Secondly, when highly connected nodes in a virus-spreading network can be easily identified, TARGET is more effective because it distributes countermeasures preemptively from one central point. However, when the topology of the virus-spreading network cannot be determined, CMC is more practical because it relies on a separate network to distribute countermeasures. This separate network can be established before the outbreak of computer viruses. Thirdly, CMC is effective when the countermeasure-propagation rate is higher than the virus-spreading rate. In the real world, this result implies that CMC is effective when the decision makers are more likely to spread countermeasures to their neighbors than to spread computer viruses, or if decision makers are more likely to discover virus infections than to stop spreading countermeasures. To achieve this goal when implementing CMC-like products or mechanisms, antivirus companies should provide incentives for customers to spread countermeasures or, alternatively, they should target customers who use computers most frequently (since they may discover the virus infection earlier than other users). Finally, CMC is most effective when the topology of the countermeasure-propagation network is such that countermeasures spread faster than computer viruses. For example, a network with a lower epidemic threshold has this property. A network having a few nodes with high connectivity exhibits this property as well. Based on this result, antivirus companies should utilize the social network of their customers (mailing lists consisting of organization representatives, for example), or set up a countermeasure-propagation network in which nodes can further spread countermeasures and some nodes are highly connected to others, similar to a peer-to-peer network[21] for distributing music files.

[21]Such as Napster (www.napster.com) and Gnutella (gnutella.wego.com).

The possible negative effects of CMC should be further studied before the strategy is completely implemented. Because countermeasures can propagate like computer viruses to cause router congestion or to deliver false information, CMC should be implemented in a way that can avoid these negative effects. Authentication processes to verify countermeasures or a death rate setting (as described in Section III-A) for slowing down the spread of countermeasures may reduce possible negative effects, but these methods need to be further studied while CMC is implemented.

Future work could be done based on our model. For example, the model and the simulation framework developed in this paper can be extended to describe a more complicated application with more states by revising the state machines. Additionally, our model simulates the spread of countermeasures and viruses through two separate complex networks. This model can be applied to other problems where there are two competing contagious agents, such as the effect of spreading rumors on the diffusion of correct information. Finally, the comparison of the four antivirus strategies has analogs to the choices for the immunization of epidemics. The current policy debate on smallpox vaccination provides a particular example for the further application of our simulation.

In summary, our approach clarifies the uncertainty of virus spreading and countermeasure propagation through different network topologies. Not only does our CMC strategy have the effectiveness that equals or exceeds the three antivirus strategies currently under consideration, but it also incorporates a richer set of variables for describing the uncertainty associated with disseminating countermeasures. In the future, we expect to further apply this network modeling approach to understand the diffusion and defenses for other classes of security incidents.

Appendix

A. An Analysis on TWL Data Set

We categorize viruses in TWL data set into two types: one-to-one and one-to-many. One-to-one refers to the virus that is designed to infect one target during one infection process that is triggered by a certain user behavior, such as a MS macro virus. One-to-many refers to the virus that is designed to infect multiple targets during one infection process that is triggered by the virus automatically, such as the Melissa virus [5] or the Love Letter virus [6]. Table V lists minimum, mean, maximum, and standard deviation of S[22] and D[23] for the two types of viruses.

From this analysis, we have found the following characteristics of computer virus spreading.

1) On average, one-to-many viruses spread faster than one-to-one viruses (the average S is higher and the average D is shorter). However, the fastest spreading one-to-one virus can spread as fast as one-to-many viruses and the variations of S between the two types

[22]Size=the number of sites that have reported a virus/ the total number of reporting sites.

[23]Duration=the last time period that a virus was shown on the list-the first time period that the virus was reported.

TABLE V
SIZE AND DURATION FROM TWL DATA SET

		all data	one-to-one	one-to-many
number of viruses		958	821	137
size (S)	minimum	0.02	0.02	0.02
	mean	0.09	0.08	0.14
	maximum	0.60	0.60	0.57
	standard deviation	0.11	0.10	0.14
duration (D)	minimum	1	1	2
	mean	15.9	16.2	15.0
	maximum	71	71	60
	standard deviation	12.6	13.1	8.5

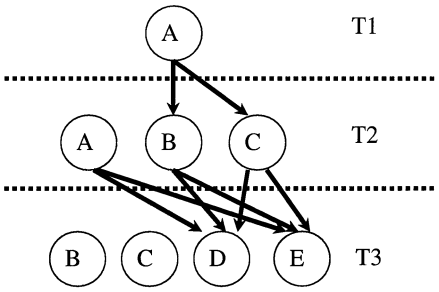

Fig. 8. Example of inferring the virus-spreading.

are similar (standard deviation $=$ 0.10 and 0.14 respectively).

2) Although the variation of D is relatively large (more than a year), we find that viruses spread much faster in the first three months of their lifetime. In the first three months of the duration of a virus, one-to-many viruses have infected an average of 83% of the total sites that are eventually infected, and one-to-one viruses have infected an average of 77% of the total infected.

B. Inferring the Virus-Spreading Network From TWL Data set

In order to infer G_v for each virus, we investigate which sites discovered that virus for each time period. We code the reporting records for each virus as a network. The data coding assumes that two reporting sites have a link to each other if one site reports a virus during the current time period and the other site reports the same virus the first time during the next time period. This assumption implies that the virus is spread from one site to another either directly from this site or indirectly through another sites during this time period. Fig. 8 shows an example that illustrates three continuous reporting periods (T_1, T_2, and T_3) for a virus. Comparing sites in T_1 and T_2, we assume that a link exists between A and B, and A and C since B and C were reported in the next time period after A was reported. We obtain G_v for this computer virus by applying the same assumption to all time periods. A similar approach to investigate the time evolution of networks has been used in social network analysis [3].

By applying this coding approach to all virus records in the TWL data set, we obtain a set of virus-spreading networks $G = \{G_k | k = 1, 2, \ldots, 958\}$. Each graph in G contains the observable nodes that a virus actually infects but does not contain the nodes that are susceptible to the virus. G_v should be larger than the observable one. For this reason, we calculate G_v as the conjunction of graphs in G, in which a link exists only if the link is observed at least in φ networks in G. In the social network analysis, this method has been used to find a central graph from a set of networks [22]. We set $\varphi = 2$ which is the largest possible network that has been used by at least two viruses. G_v calculated from this method represents the worst possible case of computer virus spreading.

REFERENCES

[1] R. M. Anderson and R. M. May, *Infectious Diseases in Humans*. Oxford, U.K.: Oxford Univ. Press, 1992.
[2] N. J. T. Bailey, *The Mathematical Theory of Infectious Diseases and Its Applications*, 2nd ed. New York: Oxford Univ. Press, 1975.
[3] D. Banks, "Metric inference for social networks," *J. Classification*, vol. 11, pp. 121–149, 1994.
[4] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, pp. 509–512, 1999.
[5] CERT/CC, *CA-99-04 Melissa Macro Virus*. Pittsburgh, PA: Carnegie Mellon Univ., Mar. 27, 1999.
[6] CERT/CC, *CA-2000-04: Love Letter Worm*. Pittsburgh, PA: Carnegie Mellon Univ., May 4, 2000.
[7] F. Cohen, *Computer Viruses*. Los Angeles, CA: Univ. Southern California, 1985.
[8] CSI, "CSI/FBI Crime and Security Survey," in *Computer Security Issues & Trends*, 2002.
[9] Z. Dezso and A.-L. Barabási, Halting Viruses in Scale-Free Networks, vol. 2002, e-print cond-mat/0107420, 2002.
[10] O. Diekmann and J. A. P. Heesterbeek, *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. New York: Wiley, 2000.
[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," presented at the ACM SIGCOMM '99 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications, Cambridge, MA, 1999.
[12] S. Gordon, "What is wild?," presented at the 20th National Information Systems Security Conference, Baltimore, MD, 1997.
[13] ICSA, Annual Computer Virus Prevalence Survey, ICSA Labs, TruSecure Corporation, Mechanicsburg, PA, 2001.
[14] J. O. Kephart and S. R. White, "Measuring and modeling computer virus prevalence," presented at the IEEE Computer Security Symp. on Research in Security and Privacy, Oakland, CA, 1993.
[15] J. O. Kephart, "How topology affects population dynamics," in *Artificial Life III*, C. G. Langton, Ed. Reading, MA: Addison-Wesley, 1994.
[16] J. O. Kephart and S. R. White, "Directed-graph epidemiological models of computer viruses," presented at the IEEE Computer Society Symp. Research in Security and Privacy, Oakland, CA, 1994.
[17] A. L. Lloyd and R. M. May, "How viruses spread among computers and people," *Science*, vol. 292, 2001.
[18] R. M. May and A. L. Lloyd, "Infection dynamics on scale-free networks," *Phys. Rev. E*, vol. 64, 2001.
[19] Y. Moreno, R. Pastor-Satorras, and A. Vespignani, "Epidemic outbreaks in complex heterogeneous networks," *Eur. Phys. J. B*, pp. 521–529, 2002.
[20] R. Pastor-Satorras and A. Vespignani, "Epidemic dynamics and endemic states in complex networks," *Phys. Rev. E*, vol. 63, 2001.
[21] ——, "Epidemics and immunization in scale-free networks," in *Handbook of Graphs and Networks: From the Genome to the Internet*, S. B. a. H. G. Schuster, Ed. Berlin, Germany: Wiley-VCH, 2002.
[22] A. Sanil, D. Banks, and K. Carley, "Models for evolving fixed node networks: model fitting and model testing," *Social Networks*, vol. 17, pp. 65–81, 1995.
[23] E. H. Spafford, "Computer viruses as artificial life," *J. Artif. Life*, 1994.
[24] C. Wang, J. C. Knight, and M. C. Elder, "On computer viral infection and the effect of immunization," in *IEEE 16th Annu. Computer Security Applications Conf.*, 2000.
[25] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1994.
[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'Small-World' networks," *Nature*, vol. 393, 1998.

Li-Chiou Chen (M'03) received the B.B.A. and M.B.A. degrees in management information systems in 1992 and 1994, respectively, from National Chengchi University, Taipei, Taiwan, R.O.C, and the Ph.D. degree in engineering and public policy from Carnegie Mellon University (CMU), Pittsburgh, PA, in 2003. Her dissertation entitled "Computational Models for Defenses against Internet-based Attacks," utilizes a network-based simulation tool to analyze the policy and economic issues in the provision of defenses against Distributed Denial of Service attacks on the Internet.

She is a Post-doctoral Research Fellow with the Center for Computational Analysis of Social and Organizational Systems, School of Computer Science, CMU. Her current research interests include complex systems modeling, pricing mechanisms for network security services and cyber insurance, vulnerability analysis of the Internet infrastructure, and security issues in online file sharing.

Dr. Chen is a member of the Association for Computing Machinery and the Association for Information Systems.

Kathleen M. Carley received two S.B. degrees, one in political science and one in economics, from the Massachusetts Institute of Technology, Cambridge, in 1978, and the Ph.D. degree in sociology from Harvard University in Cambridge, MA. in 1984.

She is a Professor of computers, organizations and society in the Institute for Software Research International, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. She is the author or co-author of five books and over 100 articles in the area of computational social and organizational science and dynamic network analysis. Recent publications include—*Designing Stress Resistant Organizations: Computational Theorizing and Crisis Applications* with Zhiang Lin (Boston, MA: Kluwer, 2003); *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers* with Ron Breiger and Pipp Pattison (Washington, DC: National Academy Press, forthcoming); *Smart Agents and Organizations of the Future in The Handbook of New Media*, edited by Leah Lievrouw & Sonia Livingstone (Thousand Oaks, CA, Sage, 2003). Her research combines cognitive science, social networks, and computer science. Her specific research areas are computational social and organization theory, group, organizational and social adaptation and evolution, dynamic network analysis, computational text analysis, and the impact of telecommunication technologies on communication and information diffusion within and among groups. Her computer simulation models meld multi-agent technology with network dynamics and are in areas such as BioWar—a city, scale model of weaponized biological attacks; OrgAhead—a model of strategic and natural organizational adaptation; and Construct—a model of the co-evolution of social and knowledge networks and personal/organizational identity and capability.

Prof. Carley is a member of the Academy of Management, Informs, International Network for Social Networks Analysis, American Sociological Society, the American Association for the Advancement of Science and Sigma XI. In 2001, she received the Lifetime Achievement Award from the Sociology & Computers Section of the ASA.