AFRL-IF-RS-TR-2003-1
Final Technical Report
January 2003

INFORMATION ASSURANCE CYBER ECOLOGY

IET, Incorporated

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J760

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-1 has been reviewed and is approved for publication.

APPROVED: 

DEBORAH A. CERINO
Project Engineer

FOR THE DIRECTOR: 

MICHAEL L. TALBERT, Maj., USAF
Technical Advisor, Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JANUARY 2003	Final Jun 00 – Jun 02

4. TITLE AND SUBTITLE
INFORMATION ASSURANCE CYBER ECOLOGY

5. FUNDING NUMBERS
C - F30602-00-C-0020
PE - 62301E
PR - IAST
TA - 00
WU - 03

6. AUTHOR(S)
Jane Jorgensen and Philippe Rossignol

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
IET, Incorporated
1911 North Fort Myer Drive
Suite 600
Arlington Virginia 22209

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency AFRL/IFTB
3701 North Fairfax Drive 525 Brooks Road
Arlington Virginia 22203-1714 Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-1

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Deborah A. Cerino/IFTB/(315) 330-1445/ Deborah.Cerino@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
Cyber Ecology is a systems-level discipline addressing the emergent properties of computer networks and their responses to perturbations, such as attacks. It is a cross-disciplinary synthesis incorporating elements of biology, epidemiology, ecology, computer science, and system engineering. In this work, methodologies from epidemiology and ecology were applied to information assurance. The goals of the Cyber Ecology project were to: (1) enable and demonstrate the discovery of noel IA technologies for the detection and mitigation of damage due to cyber attack through the application of ecological models, (2) design, develop, document, evaluate and deliver methodologies to assess the behavior of computer networks from attacks by infectious agents and direct attacks, and (3) develop and demonstrate methods to make system-level assessments about network health. The work in this report spans four major areas: (1) definition and scope of Cyber Ecology, (2) application of ecological concepts to the classification o f malicious code, in which insider threat is briefly discussed, (3) epidemiological applications of Cyber Ecology, and 94) system health expressed as emergent properties that can be assessed through evaluation of network (community) structure.

14. SUBJECT TERMS
Cyber Ecology, Cyber Epidemiology, Strategic Cyber Defense, Network Vulnerabilities, Mitigation of Cyber Attack.

15. NUMBER OF PAGES
211

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures and Tables

1 Introduction

1.1 Systems-level approach to Information Assurance (IA)

Computer networks sometimes show a remarkable resilience that belies the opportunistic way in which they are usually assembled. What is it about the structure of these networks and broader cyber systems that sometimes gives rise to resilient responses, and how can we build and manage them to make them even stronger? As ecologists, we have been drawn to study cyber systems because of the promise of finding the emergent properties, attributes that make the cyber system as a whole greater than the sum of its individual parts. While many other researchers have examined and developed specific technologies to counter cyber attack, the Cyber Ecology project has attempted to describe the underlying structure of cyber networks as dynamical systems, or cyber ecosystems, in terms of computers, users and the work they perform. We have worked to map the ecological paradigm to the cyber realm and have ported technologies that contribute to the rapid assessment of the contributions of system structure to resilience and tolerance in the face of attack. This report contains the models and tools to share this view of computer networks and cyber systems through the eyes of ecologists. Using the methodologies presented here, we assess the dynamical behavior of these systems and predict how they will respond when attacked.

The systems-level approach to information assurance has been proposed by others as well. In a position paper, Neumann (1998) aptly summarized the importance of the systems-level perspective for survivability:

> "Survivability of systems and networks is not an intrinsic low-level property of subsystems in the small. Instead it is an emergent property of entire enterprises in the large. Simply composing a system or network out [of] its components provides no certainty whatever that the resulting whole will work as desired, even if the components themselves seem to behave properly. One of the most important challenges confronting us is to be able to derive the resulting properties of a system in the large from the properties of its components and from the manner in which they are integrated."

The 'big-board' view of information assurance (IA) is an elusive goal. There is a tendency to focus on what we can measure to achieve a very precise though limited view of highly constrained, small systems, rather than addressing the more difficult-to-measure and nebulous attributes of large systems. As we increase the scale of analysis, new results emerge that pertain to the increased scale of aggregation. As the units of analysis change, corresponding to new levels of aggregation, the local effects that existed at more constrained analyses become hidden from view.

There are substantial benefits that justify such analyses and make them a valuable complement to local descriptions and experiments. The top-down, system-level view of networks provides a complementary perspective to the bottom-up mechanistic approach to structure. From the top-down, systems-level view, the aggregate structure of the network, along with its strengths and vulnerabilities, are revealed.

Examining a system from a high level of aggregation increases the size of the field of observation. This wide focus allows the capture of the direct and indirect effects of both attack and control measures on large systems. Understanding these effects in their totality permits selection of the most effective action over the widest possible solution space. This is important to ensure that control activities do not inadvertently exacerbate problematic system behaviors.

As systems grow in size and complexity, it becomes difficult to precisely control the interactions among system elements. Because management is at best distributed and at worst nonexistent, situations are likely to arise in which response to a challenge is diffuse or delayed. In computer networks, this challenge might be a distributed denial of service or other attack. Our goal is to explore the ways in which the natural tendencies of the system can be harnessed to encourage resistance to the attack and recovery to pre-attack status.

To achieve this high-level systems view of computer networks, we assume a structural approach to modeling complex systems. This provides us with more general models than can be derived from a stochastic approach because we assume that causal relationships exist based on expert knowledge rather than requiring additional evidence to infer their existence. (Of course, one component of expert knowledge is the formulation of opinions based on a deep understanding of such evidence). The resulting models cannot tell us where in the network a vulnerability lies, but can identify the entities involved.

The high-level view is important. In terms of information warfare, Alberts *et al*. (1999) noted that "[S]hared battlespace awareness emerges when all relevant elements of the warfighting ecosystem are provided with access to the COP [Common Operating Picture]. This means that battlespace awareness must be viewed as a collective property. It does not exist at just one place (node) in the battlespace, but rather at all relevant nodes in the battlespace – across echelons and functional components."

We have examined computer networks and cyber systems as ecologists and found many parallels with complex systems in the natural world. We have applied ecological and epidemiological modeling tools to discover emergent attributes that were previously hidden and have attempted to interpret these findings in an applicable and useful way. The work undertaken in this project progressed hierarchically, from the mapping of individuals, disease transmission among individuals, and community transmission of disease to analogous cyber phenomena. This report follows this chronological order in the presentation of results. It is also possible to read this report from a more applied point-of-view using the following guidelines.

The systems-level view of the network is formed by nodes and their interrelationships. The nodes may consist of collections of individuals (machines, detectors, or attackers) or aggregate variables such as a governmental department or process. The results of the analysis reveal whether or not the hierarchical structure of the network, as defined by the model, is stable, and identify points of input through which damage may be structurally maximized or minimized as the case may be. In such a vulnerability, a node represents a corridor to other nodes in the system, either through direct or indirect effects.

1.2 Technical approach

Our analysis indicates that there is no unique one-to-one correspondence between ecology and cyber ecology. Ecology, a well-developed discipline with an established body of theory, provides a starting point from which to launch our investigations. When applied to cyber ecology, ecological models suggest applications that bear further exploration. Inferences based on ecology can be mapped to testable hypotheses for cyber ecology.

The general approach for the Information Assurance Cyber ecology effort is shown in Figure 1. We based our work on the mapping of definitions and theory from ecology to cyber ecology. In the ecological domain, definitions allow for the development of theory. We will attempt a parallel transition from cyber ecological definitions to theory at each level of analysis.

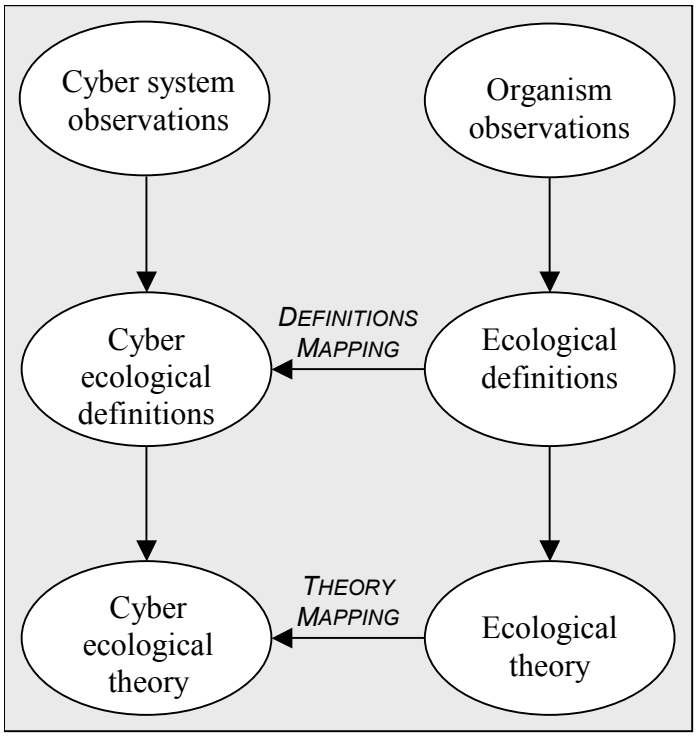


Figure 1. Technical approach for cyber ecology

In Chapter 2 of this report, we discuss ecological definitions and map these definitions to cyber ecology. We develop the experimental cyber ecological classification in Chapter 3. We review the development of ecological classification from ecological definitions in Chapter 4. We discuss the implications of ecological classification on cyber ecology in Chapter 5.

1.3 Communicating the high-level view

The methodological approaches developed in this report provide system-level analyses of infection and attack and present the results of infections and attacks to the network at the level of the network for strategic decision-making.

The synthetic view of large systems is not intuitive. Mastery of the systems-level view of network behavior can allow analysts and decision-makers to assume more strategic leadership roles when network performance is compromised by attack. These roles require an understanding of both the internal and external environments of the organization. They require the ability to incorporate ambiguity and complexity, as well as extensive information processing (Hambrick 1989).

Kotter (2001) has made the distinction that "management" is about coping with complexity. Leadership, by contrast, is about coping with change." System and network administrators deal with complexity daily, maintaining, for example, smoothly functioning computer networks. However, in times of crisis, they are also called upon to assume leadership roles to resurrect crashed systems and networks and lost data. In the event of attack, it is the administrator who is called upon to initiate actions that will disinfect the network and prevent further incursion.

Leadership itself has been described as an aggregate property of organizations. According to Jaques (1986) "leadership is not simply an idiosyncratic characteristic of some individuals. It is a systemic property derived from the interaction of the requirements of critical organizational tasks, the critical functions those tasks serve, and the problem-solving characteristics of the actors in 'leader' roles."

1.4 Anticipated impact

The approaches presented provide a framework in which to analyze system-level behavior of large networks and their response to attack. Such a framework is necessary because of the complexity of these systems. Response to perturbation can be diffuse and counterintuitive. The visualization of these systems requires the computational support and the cultivation of a strategic mindset.

Dreyfus (1982) discusses four intellectual capacities requisite for strategic thinking: component recognition, salience recognition, whole situation recognition, and decision. The first three capacities are exercised by the structural approach to ecosystem modeling. In order to construct the graphical model, the analyst must recognize that a variable represents an integral part of the system under consideration (component recognition) and that it merits inclusion in the model (salience recognition). The graphical model in its entirety forms a synthetic, holistic representation of the system and analysis reveals system-level properties (whole situation recognition). In addition, the results of the analysis inform strategic decision-making. Dreyfus (1982) also describes five stages of competency: novice, advanced beginner, competency, proficiency, and expert. As analysts become proficient in the four intellectual capacities, they advance in proficiency level. Quinn (1988) describes expert strategic leaders as master managers.

To facilitate the analysis of the models formulated in this report, we have also developed a suite of simulation programs, the Cyber Ecology Toolkit, to accompany this report. The simulation programs are written in PV Wave, a general simulation development environment. Research has shown that simulation is a viable tool for

leadership training. Streufert (Streufert 1986, Streufert *et al*. 1988) for example, described simulation-based training that produces significant increases in simulation-measured skills.

The Cyber Ecology Toolkit provides an opportunity for 'guided discovery' or mentorship in systems-level thinking and management. Future cyber terrorist attacks will be "very insidious, well-planned, highly rehearsed, and well-coordinated. . . . look for highly planned, well-researched attacks on critical pieces of information infrastructure rather than something that indiscriminately targets a wide variety of sources, for instance, a widespread denial of service attack." (Barish 2001). The effective control of such attacks will require not only the competent installation and maintenance of basic security features such as intrusion detectors, but also a more holistic, system-level perspective of the network as a whole. A system-level outlook has promise to provide system-administrators the capabilities to:

- Minimize collateral damage due to attack
- Enumerate indirect effects of attack
- Predict effects of attack
- Plan resource allocation in response to attack
- Elucidate consequences of management decisions in response to attack
- Elucidate the actual structure of cyber systems from their observed responses to attacks.

1.4.1 Anticipating 'surprise'

One issue that ecologists have addressed at length is to understand the effects of surprise on the dynamics of complex systems. Surprise may take the form of intellectual advancements that could not be anticipated with prior technology. It may also take the form of an inadequate response or malfunction of the system. In this work, we address the element of surprise in cyber systems in the form of attacks and apply techniques drawn from ecosystem analysis to describe and formulate hypotheses about the effects of these attacks.

Ecologists recognize five strategies for preparing for surprise (Levins 1995):

- Prediction
- Detection and Response
- Tolerance (Reduced Vulnerability)
- Prevention
- Mixed Strategies

We discuss each strategy briefly and indicate the section of the report in which the strategy is discussed in greater detail.

Predicting the unexpected is a contradiction in terms. The only way we can predict the unexpected is to pretend that the unknown is like the known. This biases the scope of our predictions and requires that they conform to our current understanding of the world. JBS Haldane observed, "The world is not only stranger than we imagine, but stranger than we *can* imagine." One way that we have addressed this dichotomy is by generating very general templates of attacks. For example, the general confidentiality, integrity, and availability attacks presented fit a wide range of attacks and are not constrained to any one agent in particular. The predictions presented in this report show

the hypothesized response of the cyber system as it returns to its previous homeorhetic state, that is, the resting state of the system prior to attack.

Predictions about the response to attack also play a major role in detection and response, as well as in tolerance and prevention. Hypotheses about the effects of an attack may show that certain parts of the cyber system, or network, may be more sensitive than others to the attack. These variables make good candidates for monitoring, because these variables are transparent to effects of the attack. The progress of an attack can be monitored by observing these variables.

Tolerance can also be hypothesized from the predictions about response to attack. A tolerant system will resist an attack and continue to function despite the damage rendered by the attack. Tolerant systems can be characterized by the lack of response to attack. Hypotheses about tolerance follow directly from the predictions.

Prevention occurs when predictions are proactively applied to the design and management of cyber systems. An ecological analysis of the dynamical behavior of a system prior to implementation can illustrate any weaknesses or undesirable interactions among variables prior to undertaking the expense and effort of building the system.

Mixed strategies involve the application of two or more of these primary strategies in concert. For example, different behaviors may be desirable in systems that are under attack and in those that are not. A mixed strategy might be a policy that specifies changes that should be implemented when an attack is detected in order to increase tolerance. Taken together, these strategies provide tools for the adaptive management of complex cyber systems. Adaptive environmental management of natural systems was developed in the late 1970s by Holling. He realized that "laboratory and controlled field experiments on parts of ecological systems could not be aggregated into an understanding of a whole." Adaptive management integrates science and management in a very practical way. Using Holling's insight, we treat the management of complex, changing cyber systems as experiments to be monitored and adapted.

1.4.2 Looking to the future

We have already discussed the limits of prediction and how the unknown must be described as if it were known in order to conform to our knowledge about how the world works. To gain clues about future trends, however, it is instructional to look to the fringes of current practice. What will cyber systems of the future look like?

From an ecological perspective, we are concerned with the number of species that will likely be present and the manner in which they interact. Each interaction forms a portal through which an attacker can invade a system. It is commonly believed that opportunities for infiltration will only increase as more devices and applications are drawn into the transaction stream. For example, in a recent article about the global beverage market, Stevenson (2002) described a transaction at a Japanese vending machine:

> As we watch…, a uniformed schoolgirl approaches with her DoCoMo phone at the ready. First she presses some buttons on the phone – and waits about 10 seconds. Then she holds the phone up to the scanner on the machine and waits again, for her purchase to go through. And then finally

> the machine tumbles out her selection: Water Salad, a vitamin-enhanced meal in a can that Coke targets to Japanese women.
>
> The whole routine is awkward and slightly ridiculous. If she had taken a 100 yen coin from her pocket, she would have saved at least 45 seconds: Yet this is what it takes now to sell drinks to a Japanese teenager, perhaps the most demanding breed of consumer who ever lived.
>
> A diverse cyber and communications community is required to support this one simple sale at several levels of aggregation. For the sale itself, the communications network must interact with the vending machine to perform a financial transaction. When we consider the manner in which the vending machine is stocked, the community increases in size to include a warehouse, delivery routes, and service people who perform the actual work of placing the product into the machine. Each entity involved in this transaction is a variable in the ecosystem and the way in which they interact determines the strengths and vulnerabilities of the system to stress and attack. Each interaction is a potential avenue for insertion of hostile malware.

It is true that technological advances beyond DoCoMo's i-mode are struggling in Japan's current communications technology market. The telecommunications architecture of the future may be much different that we can now imagine. However, we can expect that it will be complicated and that it will reach and interconnect markets that will in turn affect how we will live our lives. We assume the perspective that planning for attacks of the future requires more than a narrow focus on particular technologies. We must plan for broad, highly connected cyber ecosystems with complicated interaction patterns. The community concept of cyber systems is discussed in Chapter 5, and examples of cyber communities.

Ecological modeling, like most types of modeling, is an art as well as a skill. We present examples and explanations throughout this report to explain the assumptions underlying the model and the methods used to construct them. The models range in scope from very broad models at high levels of aggregation that include entire government agencies in a single variable (Propheteer Strawman), to more constrained models of process (logistics system scenario) and infectious spread (Loveletter.vbs).

A major part of the work presented concerns the mapping of ecological concepts to the cyber realm. Entities such as consumers and producers of resources and services are common to both systems, controlling the manner in which energy cascades through the respective systems. Categories such as predator and prey, parasite and parasitoid also have meaning and are discussed in Chapter 3.

We have also included epidemiological models in our work. Epidemiological models represent simplified ecological models that contain disease, hosts, and perhaps vectors. They are different from the ecological models presented here because they rely on a stochastic description of the community and produce a quantitative estimate of the force of disease. These models are discussed in Chapter 4 of this report.

1.5 Summary

We have approached the issue of cyber defense from a large-systems point of view. Using ecosystem models, we have developed methodologies to assess the effects of the attacks on cyber ecosystems.

This expansive approach addresses the scale of potential damage possible as a result of such an attack. While an attack may initially begin with penetration of a specific point in a network, the ramifications are vast, and may extend beyond the boundaries of the network-proper.

The United States has substantial information-based resources, including complex management systems and infrastructures involving control of electric power, money flow, air traffic, oil and gas, and other information-dependent items. U.S. allies and potential coalition partners are similarly increasingly dependent on various information infrastructures. Conceptually, if and when potential adversaries attempt to damage these systems using IW techniques, information warfare inevitably takes on a strategic aspect. (Molander *et al*. 1996).

Large-scale models are the domain of ecological models. By modeling information-based services and resources as elements of cyber ecosystems, we can predict the effects of attack on a larger scale, in terms of effects on the many components that form the system, than would be possible with more focused methodologies.

Future investigations might include large-scale simulations to test the speed with which the models can be generated and their accuracy (e.g., level of agreement of prediction with observed results). Using simulation experiments, we can test whether or not the models can locate weak points in a large-scale network, which are points where minimal input may potentially produce maximal damage). We can also test whether modifications guided by the analysis produce network configurations that are better able to defend critical cyber assets. The model building techniques presented in this report are just a beginning. Larger-scale, more detailed high-level scenarios will help us develop practical guidelines for analysts.

1.6 References

Alberts DS, Garstka JJ, Stein FP. 1999. *Network Centric Warfare: Developing and Leveraging Information Superiority*. CCRP Publication Series, Washington, DC. pg. 135.

Barish ST. 2001. In: Horgan D. Five thoughts about cyberterrorism Darwin 25 Oct 2001. http://www.darwinmag.com/read/thoughts/column.html?ArticleID=187).

Dreyfus SE. 1982. Formal models vs. human situational understanding: Inherent limitations on the modeling of business expertise. *Office Technology and People* 1:133-165.

Dreyfus HI, Dreyfus SE. 1986. *Mind over Machine: The power of Human Intuition and Expertise in the Era of the Computer*. Free Press, New York.

Hambrick DC (ed.), 1989. Strategic Leaders and Leadership, special issue of *Strategic Management Journal.*

Jaques E. 1986. The development of intellectual capacity. *Journal of Applied Behavioral Science* 22:361-383.

Levins R. 1995. Preparing for uncertainty. *Ecosystem Health* 1(1): 47-57.

Molander RC, Riddile AS, Andrew S, Wilson PA. 1996. Strategic Information Warfare: A New Face of War. RAND.

Neumann PG. 1998. A system-oriented perspective of survivability. *Information Survivability Workshop 1998*, Orlando FL. 28-30 October 1998.

Streufert S. 1986. How managers think and decide. *Executive Excellence* 3:7-9.

Streufert S, Pogash RM, Piasecki MT. 1988. Simulation-based assessment of managerial competence: Reliability and validity. *Personnel Psychology* 41:537-555.

Stevenson S**.** March 10, 2002. I'd like to buy the world a shelf-stable children's lactic drink. *New York Times Magazine*. Page 38.

2 Cyber Ecology

2.1 Introduction

In this chapter, we present a rationale for the application of ecological concepts to the cyber realm and demonstrate how the results of such analysis may contribute to the description, explanation, and mitigation of damage due to cyber attack.

Cyber ecology is the study of the structure and behavior of computer networks. The term *cyber ecology* is derived from ecology, the scientific study of interrelationships among organisms and their environments. Cyber ecology is a systems-level discipline addressing interrelationships among network constituents and the responses of networks to perturbations, such as attacks. It is a cross-disciplinary synthesis incorporating elements of biology, epidemiology, ecology, computer science, and systems engineering.

The application of biological concepts to computer networks has been suggested previously, notably by the National Research Council (1999) in their report, Trust in Cyberspace:

> "Metaphors and observations about the nature of our natural world – flocking birds, immunological systems, and crystalline structures in physics – might provide ideas for methods to manage networks of computers and the information they contain. The design approaches outlined above – population diversity and monitor-detect-respond – have clear analogies with biological concepts."

Ecologists study the dynamic behavior of complex natural systems and have developed techniques to study the resources and populations that form ecological communities, as well as tools to evaluate and predict the behavior of a community as a whole. While biological analogies have proven to be useful and intuitive in the description of isolated cyber phenomena, they have been applied only in an opportunistic manner. Cyber ecology is the coherent translation of biological concepts, primarily ecological, to cyber systems (see Figure 2).

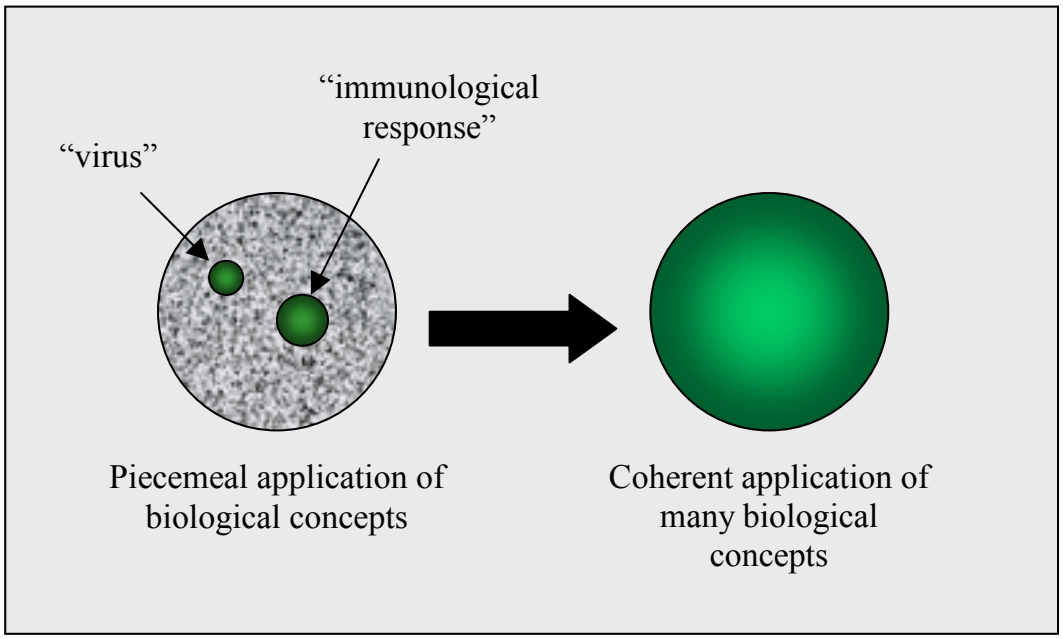


Figure 2. Cyber ecology is the coherent and consistent translation of a body of biological concepts

2.2 Role of cyber ecology in information assurance

The biological sciences are highly differentiated. Among the disciplines concerned with human health, such as medicine, public health and human ecology, there are shared areas of concern, although the sciences remain distinct. Medicine deals with the health of the individual. An underlying philosophy, *primum non nocere*, "first do no harm," illustrates the primacy of the individual. In contrast, and sometimes in short-term opposition, public health deals with the health of the population. Public health practitioners may view risk to individuals as acceptable, provided there is sufficient benefit to the population-at-large. For example, vaccination programs lower risks to the population, although a few individuals may die from adverse reactions. An ideal strategy for an individual would be to remain unvaccinated in a completely vaccinated population, thus deriving all the benefits and bearing no risk. The individual's primacy is superseded, from the standpoint of public health, by the benefits of a large proportion of vaccinated individuals in a population. Ecologists, in turn, are concerned not only about the environment in which individuals live, but also with the constitution of various populations. For an ecologist, it is important to understand the interactions among disease organisms, the individuals and populations they infect, and the environment in order to control disease. Even though they are distinct, medicine, public health, and ecology share one common goal: the reduction of morbidity (occurrence of disease) and mortality (death).

Parallels between the human disease control and information assurance models already exist. In the network sector, analogous to public health in the human disease

control model, computer epidemiology has been pioneered by Murray (1988) and Kephart, Chess, and White (1991, 1993). Parallel to the medical sector in the human disease control model, disease transmission paradigms for infectious agents in information assurance have also been suggested (Adleman 1990). IA practitioners who seek to protect individual machines through measures such as immunological response and other defensive mechanisms, function somewhat as medical practitioners in the biological model. Network administrators and IA practitioners may also intervene at a level equivalent to the level of public health practitioners in the human disease control model. They are concerned with the functioning of their networks, and may accept the loss of individuals to maintain network integrity. Cyber ecology similarly describes a specific region of information assurance (IA). Cyber ecologists are concerned with network structure and its response as a whole to perturbations such as cyber attack. All IA-related disciplines share the common goal of mitigation of damage and minimization of loss (Figure 3).

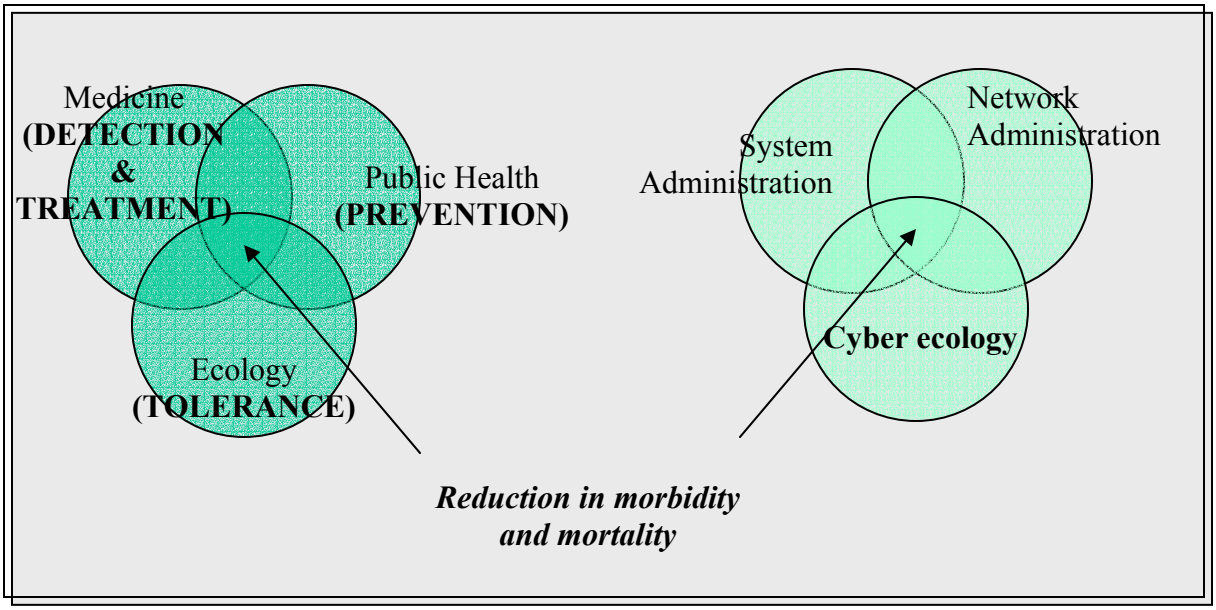


Figure 3. Parallel domains and goals of ecology and cyber ecology

Ecosystems exhibit behaviors common to all complex systems. IA seeks to understand certain behaviors that have been studied extensively by ecologists, such as:

- *Where should an attack be targeted for maximal effect?* (Ecologists in integrated pest management introduce counter-agents that maximally affect pests while minimally affecting other species and the environment.)
- *What is the most likely attack?* (Ecologists study the manner in which environmental and population changes create favorable conditions for certain pests and diseases.)
- *What are optimal counter-measures?* (Optimal counter-measures will attack root causes, not symptoms. Ecologists seek to understand the underlying mechanisms of complex system behavior manifested in the natural world.)

- *How can networks adapt to retain functionality?* (Ecosystems are constantly under attack. Ecologists investigate the reasons why some continue to function well.)
- *How can a network gracefully degrade?* (As ecosystems progress systematically through stages of succession, functionality is modified. Ecologists have explored the manner in which succession can be controlled, such as in agriculture.)
- *What redundancies contribute to robust behavior?* (Ecological models can systematically assess the contributions of redundancy, or diversity, through feedback.)
- *How can we build resilient networks?* (Ecologists have studied the manner in which ecosystems recover or fail to recover following disturbance.)

2.3 Life and its analogs

A very basic concern in the application of a biological science to a nonbiological domain is whether or not the two domains demonstrate sufficient parallel structure to support the transfer of ideas and concepts. Biological life possesses three very basic attributes: hierarchical organization, the ability to self-reproduce, and the ability to change. Organisms exchange materials and energy from the surrounding environment and transform energy in order to maintain disequilibrium with the physical forces of the local environment (e.g., gravity, heat flow, diffusion) to maintain structural integrity (Ricklefs 1990). Organisms possess a capacity for self-regulation and control, which promotes the maintenance of equilibrium. Biological communities (ecosystems) provide a backdrop of stability that allows organisms to persist and evolve through natural selection.

In contrast to previous descriptions of life expressed in terms of functionality, Maturana and Varela (1980) defined the living system as a structural and organizational unity. According to Marutana (1975), "… autopoietic systems operate as homeostatic systems that have their own organization as the critical fundamental variable that they actively maintain constant." Self-production of key organizational components is central to this concept.

Varela (1979) defined an autopoietic system formally as:

> "a network of processes of production (transformation and destruction) of components that produces components that:
> 1. through their interactions continuously regenerate and realize the network of processes (relations) that produced them; and
> 2. constitute it (the machine) as a concrete unity in the space in which they exist by specifying the topological domain of its realization as such a network."

The focus of autopoietic theory is cognition. It has not been widely accepted by ecologists, but has proven useful in models of artificial life.

Gaitlin (1972) defined life operationally as an "information processing system – a structural hierarchy of functioning units – that has acquired through evolution the ability to store and process the information necessary for its own accurate reproduction." He concurred with Shannon's (1949) suggestion that information is a capacity for storage

and transmission and that it is equivalent to the entropy of a system. The concept of information in ecology is discussed further in Section 2.10.

Farmer (in Febrache 1992) developed more detailed, criteria for artificial life, which correspond closely to the criteria for biological life:

- Life is a pattern in space-time
- Self-reproduction
- Information storage of a self-representation
- A metabolism
- Functional interactions with the environment
- Interdependence of parts
- Stability under perturbations
- Ability to evolve
- Growth or expansion

Ferbrache explained the life-like qualities of computer viruses using Farmer's criteria. They exhibit structural integrity and hierarchical organization and possess the capacity to self-reproduce and to evolve. They also exhibit self-regulatory properties that promote equilibrium states[†]. He demonstrated that computer viruses do indeed possess attributes that parallel those of living organisms. These parallel features of natural and artificial life support the application of biological models to computer networks.

2.4 Evolutionary computation

Living organisms possess the capacity to evolve. Evolutionary algorithms (genetic algorithms, evolutionary programming, evolutionary strategies, classifier systems, and genetic programming) have harnessed the biological analogy to create and maintain populations of structures that evolve using search operators (such as mutation and recombination) and rules of selection. Evolutionary algorithms comprise a set of heuristics that are valuable for solving problems whose solution has not been found using any other method.

Evolutionary defenses have been suggested as defensive strategies against computer attack (see Cohen 1993). Evolutionary defenses that provide unique defenses for each of many computers make attacks difficult by requiring a separate attack for each computer protected by such a unique defense ("security through obscurity"). Attackers would be reduced to case-by-case attacks.

It might be tempting to consider strategies such as "directed evolution," where evolution could be guided to the acquisition of known, desirable traits. This is not the nature of Darwinian evolution. In nature, evolution is a random process driven by natural selection. In evolutionary algorithms, it is a simulation driven by stochastic process. These are fundamentally dissimilar processes.

Evolutionary ideas have been applied in other areas as well. In describing the impact of the New Darwinism in economics, US Treasury Secretary Lawrence Summers (2000) succinctly summarized the impact of evolution on public policy:

[†] According to Febrache: "Stability under perturbation is a significant question which is closely related to the ability to adapt to environmental changes. A virus can modify its execution paths within tightly defined logical parameters to compensate for limited environmental changes." (Ferbrache, 1992)

> "What evolution teaches you is that improvements in innovation come in many forms. That evolution is an invisible-hand process rather than a guiding-hand process. So it inclines one toward a set of public policies that support a very dynamic and competitive economy with a lot of different people trying to do a lot of different things, rather than an approach of trying to have people in an office figuring out what's right and laying out a blueprint for the future."

This also may be the future of our information economy, the tension between IA design and adaptation strategies changing and evolving in concert with political will and public policy.

2.5 A brief history of ecology

A terse history of ecology follows. Our purpose is to highlight the emergence of three methodological approaches: thermodynamic, cybernetic and evolutionary.

Ecology began as a descriptive science. The first ecologists were natural historians. These early naturalists in the 18th and 19th centuries were concerned with the balance of nature, believing it to be God's plan. Charles Darwin (1859) shattered these beliefs with the publication of his theories in *On the Origin of Species*. Darwin's theory states that complex interactions among species play an integral role in the formation and maintenance of this balance.

Many early ecologists assumed a strictly biological interpretation of natural communities. Early American ecology was strongly influenced by F. E. Clements (1916) and the Clementsian school of organismic ecology. In this early systems-level conceptualization, communities of organisms were defined as super-organisms. Elton (1927) pioneered the concepts of trophic levels (i.e., feeding levels, such as producers, herbivores and predators) and food webs, tracking the unidirectional flow of energy through the community.

Clements (1936) and H.C. Cowles (1899) developed the principle of ecological succession, which documented that communities change over time in a cyclic manner. New species colonize disturbed habitat, initiating the process. These species may modify the environment, allowing later species to establish themselves, or conversely, to prevent new species from entering the system through competition.

Incorporation of the environment into the community is relatively recent. A.G. Tansley (1935) coined the term "ecosystem" to capture both the biotic (living) and abiotic (non-living) elements of the community. Raymond Lindeman (1942) built upon this construct by distinguishing between the one-way flow of energy and cycling of chemical substances in the ecosystem. E.P. Odum (1953) continued this work by developing complex models of energy flow through ecosystems.

The thermodynamic approach to ecology was contributed by A.J. Lotka in 1925. His models demonstrated that transformations of mass and energy in ecosystems conformed to thermodynamic laws and described ecosystems mathematically in terms of the interactions among constituents of the community.

Systems ecologists continue to describe complex ecosystems using mathematical models. The cybernetic approach to ecosystem analysis emphasizes the roles of feedback and control in determining the relative abundances of the constituents of the community.

The study of ecosystems as complex, dynamic processes lends unique insights into many apparently paradoxical system responses. In his classic investigation of the role of predators in determining the overall structure of a community, Paine (1966) demonstrated that certain predators, termed "keystone", are necessary components of ecosystems with two or more competing prey species. These keystone predators stabilize the system by preying on both competing prey species, allowing them to coexist within the limited resources of the system. Removal of a keystone predator causes the community to collapse as competing prey species drive each other to extinction. Paine's work also marked the beginning of experimental community ecology.

Modern ecology has been formed through the incorporation of two new concepts: equilibrium and evolution. Communities may remain structurally unchanged over time although energy and nutrients pass through them and organisms die and are replaced. They are resistant to perturbation and return to equilibrium following such a disturbance.

Evolution, at its most basic level, refers to change through time. Biologically, it refers to the process of speciation, the change that species, or reproductively distinct groups, undergo through time. Darwinian evolution arises from mutations, namely random changes in the highly conservative genetic code. An individual with a mutation will have a higher, lower or equal fitness to other members of the population. If a mutation confers higher fitness, it will be "naturally selected" and eventually become dominant. It is important to realize that any selection, natural or otherwise, requires a stable equilibrium against which the fitness of traits can be compared. When natural selection occurs, equilibrium provides a necessary backdrop against which evolution may occur.

Ecology has progressed through the stages of natural history (what organisms are present?), population ecology (how do the organisms interact?) and community ecology (what is the system-level response to change?). Given this progression, cyber ecology is currently in the early natural history stage. The *ad hoc* classification of attack agents using biological analogy, such as viruses and worms, resembles the work of early ecologists describing the life history of organisms. Our goal has been to develop cyber ecology through its subsequent stages, exploring the complex, dynamic nature of computer users and attackers within the structure of computer networks.

2.6 How do ecologists build models?

A scientist's perspectives of the natural world are determined by the problems studied. In some disciplines, it may be appropriate to focus on molecular structures, while in others a more macroscopic view may be necessary. In his classic paper, Levins (1966) described the process of model building in ecology as managing tradeoffs among generality, realism and precision. He listed three model-building strategies:

(1) sacrifice generality for realism and precision.

These models yield precise predictions for tightly constrained situations. This approach has been adopted by natural resource managers to formulate precise, testable predictions based on the short-term behavior of organisms.

(2) sacrifice realism for generality and precision.

Popular among physicists who enter ecology, this approach yields very general models that generate very precise predictions. However, the equations are

unrealistic given the conditions in the natural world. Small departures from initial assumptions often have large effects upon predicted outcome.

(3) sacrifice precision for generality and realism.

Using qualitative (monotonic) data, realistic and generalizable predictions of system behavior can be hypothesized and tested. This approach is valuable for hypothesis generation. It is the most practical approach of the three in that data may be collected quickly and inexpensively.

Data differ in the types of information they contain. When we speak of precision, we usually refer to ratio data, those data in which the ratio between two quantities has meaning. Interval and ordinal data also contain quantitative information, but at a coarser scale. For interval data, distance (far or near) has meaning, but ratios have none. For ordinal data, direction has meaning (higher or lower), but distance and ratio do not. In each category of data, quality is an issue. We would argue that ordinal data of high quality is more valuable than erroneous ratio data. In Table 1, we describe types of data, their characteristics and the analyses to which specific types of data are applied.

Table 1. Data types and their uses

Data Type	Characteristics	Analyses
Nominal	Classification by name	Taxonomic
Ordinal	Order has meaning	Nonparametric statistics Qualitative analysis
Interval	Distance has meaning	Nonparametric statistics
Ratio	Ratios have meaning	Parametric statistics Quantitative analyses

Generalizability will be more important than precision in our initial hypotheses about system behavior. Since trends will be more important than point estimates, ordinal data describing the direction of effect will be sufficient. As our hypotheses become more refined, we will turn to more quantitatively dense data. Until that time, we will be more interested in the efficacy of structural modification and input in describing changes in system response rather than in efficiency. That is, we will focus on the general potential of changes in structure to influence system behavior, rather than on the magnitudes of specific changes in specific situations.

2.7 What are the structural elements of ecological models?

Our models will be based on the construct of the community, an association of interacting populations. The community is formed by a rich hierarchy of processes that are carried out over different scales in space and time. It is a dynamic entity formed by the continual flux of resources through the system and by the birth, death, and growth of individuals. The populations forming a community are linked so that the ecological impact of a population (its influence on predators, competitors and prey) extends throughout the system.

Ecosystems are open systems, based on the ability of producers to transform energy taken from outside the system into a consumable form. Producers take the energy from the sun and through the process of photosynthesis, convert this energy into food.

Ecosystem models contain variables. These variables must be measurable and, of course, variable. Often, they are species populations. They may also be nutrients, such as nitrogen, if the levels of the nutrient vary. The variables must be linked. These links, not the variables, contain the fundamental structural information about the ecosystem. They represent gates or conduits through which energy exchange occurs.

The interrelationships among variables in a community are described as dyads. From a qualitative point of view, variables have two possible direct effects upon each other: an increase in one variable may cause an increase (positive effect) in another, or it may cause a decrease (negative effect). In a predator-prey relationship, the predator can cause the prey population to decrease while the prey population causes the predator population to decrease. In a competitive relationship, each population can cause a decrease in the other. In a mutualistic relationship, each species can cause an increase in the other. Interrelationships need not be reciprocal. The possible direct links between variables are described in Table 2. (Variables that are not directly linked may still influence each other indirectly through feedback loops.) Strong interrelationships among system components are not necessarily required for stability. McCann *et al*. (1998) have shown that weak interrelationships can be stabilizing.

Table 2. Possible interactions between species (from Stiling, 1996)

Nature of Interaction	Species 1	Species 2
Mutualism	+	+
Commensalism	+	0
Herbivory, Predator/prey	+`	-
Parasitism	-	-
Allelopathy	-	0
Competition	-	-

Ecosystem models need not be complex. The simplest possible ecosystem model is shown in Figure 4. It consists of a single self-regulated population connected to its environment by a double link. It must both affect and be affected by the other populations and resources in the larger system. Of course, it is possible to complicate the model. After all, most life on earth derives its energy from the sun, and the simple model can be made complicated by tracing energy to this initial source. However, this may not be practical or necessarily enlightening.

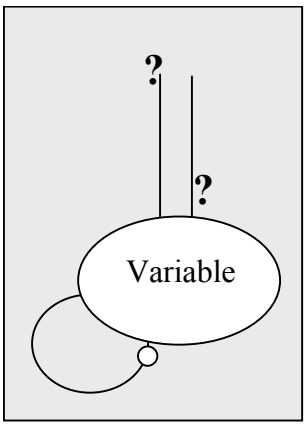


Figure 4. Simplest possible ecosystem:

A self-regulated variable linked to the outside world by a double link

Density dependence and links to allochthonous (i.e., input from outside the system) input are represented as self-effects. The presence of a strong regenerative effect in at least one variable, expressed as a negative self-effect, is a requirement for system stability. Intuitively, this requirement states that in a stable system, the most basic variables, upon which all of the more complex interrelationships are formed, must be able to maintain themselves independently of the rest of the system.

Communities can be represented mathematically by the community matrix. Just as information describing interrelationships is contained in the links, information describing system behavior is contained in the characteristic polynomial of the community matrix. The coefficients of the characteristic polynomial express the role of feedback in the system. Feedback occurs in ecosystems as the effects of interrelationships cycle through the community. Systems can be loosely categorized according to level of feedback:

- *Ad hoc* assemblage (no feedback)
- Controlled interactions (humans exploit biological tendency; e.g., agriculture)
- Evolving, self-regulatory, autonomous communities.

Feedback is important to the ecosystem because is provides regulation and stability. A stable system is a predictable one, and a predictable system can be managed. The solution to the characteristic polynomial, the eigenvalues, describe system recovery following perturbation.

2.8 Structure in ecosystems

Systems are portrayed as concrete assemblages of things, but actually are abstract constructs. Systems are selectively envisioned to elucidate specific phenomena. The challenge is to build informative models that incorporate sufficient complexity.

2.8.1 Overview

Ecologists refine their analyses of communities by the use of different structural perspectives. Food web analysis traces the flow of energy through a community by noting who eats whom. Variables are usually species populations.

Ecosystem analysis, in which species are placed in functional groups with other species of similar trophic position, is another way to visualize communities. Ecosystem analyses tend to be coarser than food web analyses, but may capture more basic elements of community structure by reducing the number of variables in the ecosystem. It is important to bear in mind, however, that differences in the underlying approach taken in formulating these models (e.g., thermodynamic or cybernetic) can drastically affect predictions generated by the model.

Biologists and ecologists use parsimony to distinguish among models. Parsimony is the philosophical concept that states that of two or more explanations, the simplest should be accepted. It is often referred to as "Occam's razor," from the medieval scholastic philosopher who formalized it. In evolution, it is invoked in the science of systematics, which is the reconstruction of evolutionary paths. When faced with two paths, for example, the shortest is more likely because mutations are extremely rare and successful ones even rarer. It is therefore very unlikely that the longer path could have occurred.

In cyber ecological models, food web analysis is analogous to tracking the flow of information thorough a computer network as machines communicate. Information transfer from a server to client can be thought of as a trophic relationship. Resources from the server (for example, information and bandwidth) are "consumed" by the client. Functional groups may consist of types of users (end-users, administrators) or operating systems (Mac, UNIX, PC). Rather than including many distinct nodes in a network model, we can collapse the number of variables significantly by using functional groupings. This exercise may or may not be informative depending on the underlying structure of the network.

2.8.2 Hierarchy

The modern view of hierarchy is that any system may be viewed simultaneously at a range of scales as a multilayered composite (Peters 1995). This view is intuitive for biologists who are familiar with the concept of "levels of organization" (cell, organism, population, community). In ecosystem analysis, this multilayer perspective applies as well. For example, plant populations may be modeled at many levels, in terms of their nutrient value to primary consumers at one level and in terms of their chemical exchange processes at a finer level.

Simon (1962, 1969, 1972) introduced the insight that organization results from differences in process rates. Overton (1974) showed that the structure imposed by these differences was sufficient to decompose a system into organizational levels. Systems may be modeled as vertical structures where behaviors associated with higher levels occur at slow rates while those associated with lower levels occur at rapid rates. For example, the photosynthetic processes in the leaves of a tree occur at a more rapid rate relative to its growth. The growth of a forest is a slower process at an higher level of organization. In the vertical view of hierarchy, organizational levels are isolated from each other because they operate at distinctly different rates.

The differences in rates help explain how ecosystems respond to environmental fluctuations. Each level in the hierarchy filters the signals it receives by attenuating those greater than its own characteristic frequency. Each organizational level acts as a filter, confining high-frequency dynamics to lower organizational levels (Overton 1977). Lower organizational levels communicate an averaged, filtered response to higher levels.

Ecosystems also may be decomposed horizontally into subsystems or holons (Koestler 1967, 1969) on the basis of differences among rates. Within a holon, system components interact strongly with each other. They interact weakly with components of other holons. Holons are defined by an enclosing boundary or surface that encloses it and separates it from the rest of the system (T.F.H. Allen *et al*. 1984).

The definition of an ecosystem is not arbitrary, but dependent upon the time and spatial scale of the problem being addressed. The level of organization determines the breadth of observation. For example, higher-level behaviors, such as the growth of a forest, may occur slowly and adequate lengths of time must be allocated to observe change. When observing lower-level behaviors, these same higher level behaviors will appear as background constants. The scale of observation depends on the "window (i.e., the range of rates) through which one is viewing the natural world" (O'Neill *et al*. 1986).

O'Neill *et al*. (1986) noted that it is impossible to designate *the* components of *the* ecosystem. A group of trees in a forest may appear:
(1) as a dynamic entity in its own right;
(2) as a constant (i.e., nondynamic) background within which an organism operates; or
(3) as inconsequential noise in major geomorphological processes.
Choice of the appropriate spatio-temporal scale is of paramount importance.

It may be tempting to collect data about lower-level behaviors with the intention of picking out patterns of higher-level behaviors. This is more often than not a costly exercise in futility. The art of modeling is the choice of the appropriate window through which to observe a behavior of interest. Massive amounts of data at too fine a scale will obscure the detection of long-term patterns. However, the converse is also true. Too few data about lower-level behaviors also will be non-informative.

2.9 Reductionism and emergence

Ecosystems consist of many parts interacting in complex ways. Simon (1962) noted that while it is possible to elucidate the behavior of the parts of a complex ecosystem, it is more difficult to understand the properties of the whole. Reductionist examinations are easier than synthetic analyses of complex systems, but cannot, by definition, capture the emergent properties of the system. "A reductive explanation of a behavior or a property of a system is one showing it to be mechanistically explicable in terms of the properties of and interactions among the parts of the system" (Wimsatt 1997). Emergent properties are dependent upon the mode of organization of the system's parts. That is, they involve the interorganizational interdependence of diverse parts.

Emergent systems not aggregates. Aggregates demonstrate invariance to (1) rearrangement of parts; (2) addition or subtraction of parts; (3) decomposition and reaggregation of parts; and (4) cooperative or inhibitory interactions. An assemblage is "merely" an aggregate when all conditions are met for all possible decompositions of the system into parts. Emergence is the manifestation of new properties when a system is perturbed. Sometimes a system may behave as an aggregate under one decomposition,

but display emergence on others. "The common appearance of unqualified aggregativity is a chimera" (Wimsatt 1997). The multiple perspectives assumed by ecologists tease apart these paradoxical decompositions to construct compositional models of complex systems.

2.10 Information in ecology

One quality of life is the ability to remain at disequilibrium with the immediate environment, so that individuals may maintain life-sustaining processes without succumbing to the powerful equalizing forces of the surrounding environment. For example, organisms must sustain internal chemical processes such as the balance of electrolytic chemicals exclusively from chemical processes occurring in the external environment. The primary producers harness the energy of the sun through photosynthesis and provide the basis for most of the food chain. Consumption is a repeated process of transforming matter to energy to matter. Information is associated with these state changes. A balance results between the tendency toward a more uniform distribution of energy of lower quality (the level of energy in the immediate environment) with increasing complexity of traces left by decay of energy. Ecosystems are a mechanism for organizing huge amounts of matter. This mechanism is driven by energy transitions that occur as energy flows through and is captured by the system.

The cybernetic organization of ecosystems arises from the concept of the ecosystem as a self-controlled entity depending on a network of information exchanges and negative feedback. System properties are maintained by feedback resulting from energy-matter transfers and information exchange.

Shannon (1949) defined information as a quantitative measure of communicative exchange, as the capacity to transmit information, or as Gaitlin (1972) has suggested "potential information." As Weaver (1949) noted, ". . . this word 'information' in communication theory relates not so much to what you do say, as to what you could say." While most biologists have focused on mechanical descriptions of life, for example, in terms of genetic and chemical processes, some have also explored life in terms of information processing.

The incorporation of information theory into ecology has been pursued extensively by Margalef (1968, 1995). Many of the views that follow about ecology and information theory in this section are summarized from his work. Margalef considers ecosystems and communication to be very similar. Both ecosystems and language belong to a class of systems made of parts (subsystems) that are self-replicating by their own power (biological organisms) or by external agency (viruses, words). The ecosystem acts as a channel in which the relationship of components, of each single individual to another, may be established.

Margalef sees the ecosystem as information, whose maintenance and accretion involves cybernetic behavior. "Any cybernetic system, through the interaction of its parts, restricts the immensely large number of a priori possible states and, in consequence, carries information" (Margalef 1968). In ecosystems, loss of energy quality is associated with increased information and entropy. The largest increase in entropy occurs in primary producers, the site of a major ecological information exchange. Energy decay corresponds to increased information.

The accumulation of information in ecosystems, according to Margalef (1968), is accomplished through succession. Self-organizing systems pass through states in which some piece of the system is replaced by some other piece that allows the preservation of the same amount of information at lower cost. In initial poorly organized stages of succession, where exploitative species colonize pristine landscapes, the relative energy flow is high as organisms receive the full impact of the environment and are selectively destroyed. Information accretion is fed by the surplus production of new organisms representing the cost of accumulating information.

Ulanowicz (1980, 1995, Hirata & Ulanowicz 1984) has used information theoretic concepts to quantify the growth and development of ecosystems. His metric, *ascendency*, is based upon mutual information, the average amount of uncertainty resolved by knowledge of network structure. Ascendency is an attribute of the ecosystem as a whole. As ecosystems "mature," underlying transformations tend to contribute to higher network ascendency.

2.10.1 Information storage (memory)

Feedback loops formed by the interaction of species are a form of memory for ecosystems. They are expensive to maintain and only contain a limited capacity for storage. For example, the information content of contemporary forests, contained in their feedback loops, does not differ significantly from the information content of ancient forests. Margalef (1968) suggests that ecological memory "seems always to have played the role of an auxiliary memory of rather limited capacity."

In cybernetic systems where information is expressed by the actions of mechanism, storing information means increasing the complexity of mechanism. One reason for the success of life is in increasing complexity through miniaturization, "packing, in a small space, a prodigious number of overlapping mechanisms, wonderfully persistent by virtue of built in regulatory circuits and sufficiently open to carry into the future a promise of new developments" (Margalef 1968).

Information can also be stored in persistent structures. These structures may consist of large animals and trees with prodigious life spans. Information is also stored by species that construct edifices and artifacts.

2.10.2 Ecosystem as bridge

Margalef (1995) considered the ecosystem to be a bridge of information. "If energy provides a bridge over space, information assures a persistence, or marks an evolution, that is expressed along time." Information is transmitted, in Margalef's view along three subchannels: genetic, ecological, and cultural. The genetic subchannel contains information about replicable individual structures. The ecological channel contains information about constituent species. The cultural channel contains information that has been learned by individual activity or experience (Margelef 1968, 1995).

Throughout the development of life on earth, the cultural channel has been of negligible importance until only recently when it has been subject to explosive growth. In ancient times, the cultural channel was expressed through primitive signals such as the formation of trails that others could follow and accumulations of dead material. It progressed to more complicated manifestations, such as the formation of local traditions.

It is now expressed on a large scale in our expansive manipulation of the environment (and associated untoward effects such as global warming), and on a very small scale in our manipulation of the bits and bytes that form our vast computational networks.

2.11 Experimental community ecology

Experimental ecology is a relatively recent development. Analytical food web analysis began in the mid-1950s, sparked by the question of whether or not community diversity enhances stability. MacArthur (1955) suggested that the more complex a community, the greater its stability, stimulating experimental work to address this issue. He reasoned that in systems where energy can take many paths, a disruption of one pathway, say the unavailability of one prey species, can result in the diversion of energy through another route, say an alternative prey species, maintaining the overall flow of energy through the system.

Ecological experiments have been criticized because of their inherent ambiguity (Peters 1991). As we have previously discussed, many perspectives of the ecosystem are simultaneously possible, and models generated from these views may differ in their predictions. The ways of describing structure, hierarchically as in food web analysis or grouped into functional groups, are often too flexible for those seeking precise numerical results. We should keep in mind, however, that the goal of complex ecosystem analysis is to learn about structure. Often, apparently contradictory results are very consistent when viewed from the perspective of the community as a whole.

The key to community structure lies in the links among variables, and experimental community ecology seeks to gain information about how variables are connected and how these connections affect system response. Experimental community ecology is reductionist in that it teases apart the individual components of system behavior, yet it is synthetic in that this information must be recombined into a description of the system to predict system level response.

Community ecological experiments can be arranged according to the amount of knowledge required to perform them. Broadly speaking, experiments in which ecosystems are manipulated are called *pulses* and *presses*. In a pulse experiment, a variable is pushed from its current state, for example, the number of prey may be increased, and the system's return to its previous state is observed. In a press experiment, a permanent change is made to the strength of a link. Frequent exploratory presses can be used to monitor system structure and state. Changes in the interrelationships among system components will be evident through the results of press experiments. The knowledge required for ecological experimentation to verify predictions about system structure is summarized in Table 3.

Table 3. Knowledge required for ecological experimentation to verify predictions about system structure

KNOWLEDGE REQUIRED ↑	Knowledge about system	Observation/Experiments
	Explicit knowledge of system; manipulate parameters to evaluate effect of change on system-level response (ecological intervention experiments)	Pulse experiment
		Add/remove a link
		Add a variable
		Remove a variable
		Press experiment
	System well understood; evaluate effects of known parameter change	Observation of effects of natural experiments (floods, fire, meteor strikes) - natural presses
	System not well understood; assess patterns of correlation to infer parameter change	Observation of patterns of correlation from historical record

Large-scale ecosystem experiments are impossible to replicate because of the size of the experimental unit and uniqueness of experimental conditions. Ecologists look at natural presses and pulses, such as meteor strikes, eclipses, floods and weather phenomena (such as El Niño and La Niña) to observe if system response matches analytical predictions. The uniqueness of the experimental unit has also prompted some ecologists to question the underlying stochastic methodology used to analyze such natural experiments. Some analysts assert that ecologists are in fact Bayesians rather than frequentists (Ellison 1996).

Frequentists believe that there is a true, fixed value for each parameter of interest and that the expected value of this parameter is the average value obtained through repeated random sampling. Bayesians, in contrast believe that statistical parameters are random variables. Bayesian inference evaluates the probability that an explicit scientific hypothesis is true given ("conditioned on") a set of data. Bayesian inference incorporates the analyst's beliefs about data as prior probabilities before an experiment is performed, and incorporates new knowledge from the experiment to derive a posterior probability. The process is iterated to refine parameter estimates (Ellison 1996). Bayesian statistics present an elegant method for dealing with uncertainty, particularly when replicates (such as ecosystems or large computer networks) are rare or nonexistent.

Cyber ecosystems are complex systems that exhibit hierarchical structure. Individual computers link together to form LANs which then are connected into larger networks. Although the temporal scale of observation for some system components may be considerably reduced, to the level of nanoseconds and smaller, humans are still an integral part of the system and are associated with much longer response times. Cyber ecosystems may also be stratified "spatially", for example by operating system platform (PC, Unix, Mac) or by access to specialized networks (NIPRnet, SIPRnet, Internet). The multi-layer hierarchical perspective may prove valuable to capture the many simultaneous aspects of these systems.

2.12 Generating hypotheses for cyber ecology

Since cyber ecology is a new field, new hypotheses may be as enlightening as results. We face foundational issues. What is a variable? What is the unit of measurement for the strength of a link? Other than through raw communication protocols, how are computers connected into networks?

In order to obtain broad perspectives about the structure and system-level responses of computer networks in this early stage of study, we will use Levins's third modeling strategy: sacrifice precision for generality and realism. Once we derive models that circumscribe broad classes of behavior, we will be able to perform more detailed studies in identified areas. Initial descriptions and predictions of system-level response can be made and revised quickly with input from subject matter experts who will not need to invest heavily in the modeling technology.

Levins's loop analysis provides a framework for using qualitative (monotonic data) to develop mathematically rigorous predictions of system behavior. Developed from concepts of qualitative equilibrium in engineering (Mason 1953) and economics (Quirk and Rupert 1955), the technique assesses the compliance of system parameters with the Routh-Hurwitz criteria for stability. Briefly stated, these criteria state that feedback at all levels must be negative and that higher level feedback must be less than lower level feedback. Intuitively, these criteria make sense. A stable system must take the time and resources to renew itself or it will collapse. Loop analysis for mid- to large-scale systems was impractical until only recently with the advent of powerful symbolic processors for personal computers.

Loop analysis has been criticized because of the ambiguity of its predictions. Any interconnected community consisting of three or more components will be conditionally stable. That is, any one of a myriad of point solutions will satisfy the conditions of local equilibrium. Recent advances for assessing the determinacy of the predictions, in which weak predictions can be identified, have rectified many of these problems (Dambacher 2000).

Loop analysis supports modeling by providing a means for constructing and testing models quickly and inexpensively (see Chapter 5 of this report). Analysis of the community matrix gives the conditional stability criteria for a system. The predictions obtained from loop analysis yield the direction of system response to a specific input.

2.13 Humans in the loop

Evidence of human impact on the environment is everywhere. There is no spot on earth that remains untouched, either by the insidious spread of chemical pollutants, the effects of massive civil engineering projects (such as dams) climatic change, or deforestation. In modeling the environment, man must be considered, either as an active participant within the system or as a source of input from outside the system.

Human intervention adds the dimension of intent. Above all else, humans depend on the environment for survival. However, rather than merely coping with the effects of environmental change, humans seek to actively manage nature, selectively creating and destroying parts and wholes of ecosystems to satisfy human goals. These actions may run counter to any logical pattern we may hope to discern in a self-organizing system.

Cyber systems may resemble agricultural systems more closely than freely evolving ecosystems. Agriculture is one example of human exploitation of the

environment. The system is simplified to include fewer species and the cycling of resources into the system becomes more important. The cyclicity of the system is emphasized as its maturity is held in check. Exploitation exerts a rejuvenating effect on exploited ecosystems that we may also observe in cyber ecosystems.

2.13.1 Science is not enough

Humans are political beings. Often what is scientifically obvious is not politically expedient. Tradeoffs must be evaluated and decisions are often made that are inconsistent with best science. Ecologists have felt the effects of the political dilution of science keenly. For example, although many ecologists have found that habitat restoration for stocks of endangered fish, such as wild salmon, can only be accomplished through removal of hydroelectric dams, the demand for power makes this choice politically unfeasible.

Ecologists have developed methods for incorporating stakeholder values and opinions into natural resource management. Although, depending on one's point of view, these have had limited success in natural resources management, they may prove valuable as tradeoffs become apparent in the implementation of IA measures. In the adaptive management paradigm (Walters 1986, Walters and Holling 1990), for example, management is viewed as an iterative experiment where stakeholders reach consensus about management alternatives. The process can be difficult and lengthy, but does result in the incorporation of best science into natural resources management.

2.14 Summary

In this chapter, we have explored the heritage of cyber ecology and the application of ecological theory and analytical techniques to this new domain. We presented the following broad conclusions:

- The development of cyber ecology is consistent with the historical development of ecology, albeit at an accelerated pace.
- Structural similarities between ecological communities and computer networks suggest that the transfer of domain knowledge is justified.
- System behavior of the whole is the expression of countervailing forces among components. A synthetic approach is required to understand the apparently paradoxical behavior of complex, dynamic systems such as ecosystems and computer networks.
- The thermodynamic, cybernetic and evolutionary perspectives of ecology also apply to cyber ecology. The cybernetic approach is perhaps the most interesting in that the internal regulatory behavior of a network can be modified to achieve a desired system-level response (or resistance to response).
- Experimental techniques used by ecologists show promise in elucidating the structure of computer networks. In particular it may be possible to manipulate these systems experimentally. Frequent exploratory interventions may represent a strategic tool for dynamically modeling changes in system structure.

2.15 References

Albert R, Jeong H, Barabasi A-L. 2000. Error and attack tolerance of complex networks. *Nature*. 406:378-381.

Allen TFH, O'Neill RV, and Hoeskstra TW. 1984. Interlevel relations in ecological research and management: some working principles from hierarchy theory. General Technical Report RM-110, United States Department of Agriculture, Rocky Mountain Forest and Range Experiment Station, Fort Collins, CO.

Barabasi A-L, Albert R. 1999. Emergence of scaling in random networks. *Science*. 286:509-512.

Bosserman RW. 1979. The hierarchical integrity of *Utricula*-Periphyton microecosystems. Ph.D. diss., Univ. of Georgia, Athens.

Bush SF, Hershey J, Vosburgh K. 2000. Brittle system analysis. *Proceedings of Vwsim '00: Virtual Worlds and Simulation Conference, WMC '00*. 2000 SCS Western Multi-Conference, San Diego.

Clements FE. 1916. *Plant Succession: An Analysis of the Development of Vegetation*. Carnegie Inst. Wash. Publ. 242.

Clements FE. 1936. Nature and structure of the climax. *J. Ecol*. 24:252-284.

Cohen F. 1989. Models of practical defenses against computer viruses. *Computers and Security*. 8; 149-152.

Cowles HC. 1899. The ecological relations of the vegetation in the sand dunes of Lake Michigan. *Bot . Gaz*. 27:95-117, 167-202, 281-308, 361-391.

Dambacher JD, Li HW, Rossignol PA. 2002. Relevance of community structure in assessing the indeterminacy of ecological predictions. *Ecology*.

Darwin C. 1859. *On the Origin of Species*. Murray, London.

Ellison AM. 1996. An introduction to Bayesian inference for ecological research and environmental decision-making. *Ecological Applications*. 6(4):1036-1046.

Elton CS. 1927. *Animal Ecology*. Macmillan, New York.

Faloutsos M, Faloutsos P, Faloutsos C. 1999. On power-law relationships of the Internet topology. *SIGCOMM 99*. ACM, Cambridge, MA

Ferbrache D. 1992. *A Pathology of Computer Viruses*. Springer-Verlag, London.

Gatlin LL. 1972. *Information Theory and the Living System*. Columbia University Press, New York.

Huberman BA, Adamic LA. 1999. Growth dynamics of the World-Wide Web. *Nature*. 401:131.

Harte J. 1979. Ecosystem stability and the distribution of community matrix eigenvalues. In: Halfon, E., ed. *Theoretical Systems Ecology*. Academic Press, New York. 453-465.

Hirata H, Ulanowicz RE. 1984. Information theoretical analysis of ecological networks. *Int. J. Systems Sci*. 15: 261-270.

Holling CS. 1973. Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics*. 4:1-23.

Holling CS, Clark WC. 1975. Notes towards a science of ecological management. In: Dobben, W. H. and R. H. Lowe-McMonnell, eds. *Unifying Concepts in Ecology: Report of the Plenary Sessions of the First International Congress of Ecology*. Dr. W. Junk BV Publishers, The Hague, the Netherlands.

Janssen MA, Carpenter SR. 1999. Managing the resilience of lakes: a multi-agent modeling approach. *Conservation Ecol*. 3(2):15.

Kephart JO, Chess, DM, White SR. 1993. Computers and Epidemiology. *IEEE Spectrum*, May, 1993.

Kephart JO, White SR. 1991. Directed-graph epidemiological models of computer viruses. *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy; Oakland, California, May 20-22, 1991*: 343-359.

Kephart JO, White SR. 1993. Measuring and modeling computer virus prevalence. *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy; Oakland, California, May 24-26, 1993*: 2-15.

Koestler A. 1967. *The Ghost in the Machine*. Macmillan, New York.

Koestler A. 1969. Beyond atomism and holism – the concept of the holon. In: Koestler, A. and Smythies, J.R.(eds.) *Beyond Reductionism*. Hutchinson, London.

Levins R. 1966. The strategy of model building in population biology. *American Scientist*. 54:421-431.

Lindema, R L. 1942. The trophic dynamic aspect of ecology. *Ecol*ogy 23:399-418.

Lotka AJ. 1925. *Elements of Physical Biology*. Williams and Wilkins, Baltimore.

MacArthur RH. 1955. Fluctuations of animal populations and a measure of community stability. *Ecol*. 36:533-536.

Margalef R. 1968. *Perspectives in Ecological Theory*. Univ. of Chicago Press, Chicago.
MargalefR. 1995. Information Theory and Complex Ecology. In Patten BC, Jorgensen SE eds. *Complex Ecology: The Part-Whole Relation in Ecosystems*. Prentice Hall, Englewood Cliffs, New Jersey.

Marutana HR. 1975. The organization of the living: a theory of the living organization. *International Journal of Man-Machine Studies*. 7:313-332.

Marutana H, Varela, F. 1980. Autopoiesis and cognition: the realization of the living. In: Cohen, R.S. and Eartofsky, M.W. (eds*.). Boston Studies in the Philosophy of Science*, Vol 42. D. Reidel Publishing Co., Dordecht.

Mason SJ. 1953. Feedback theory – some properties of signal flow graphs. *Proceedings of the Institute of Radio Engineers*. 41:1144-1156.

McCann K, Hastings A, Huxel GR. 1998. Weak trophic interactions and the balance of nature. *Nature*. 395:794-797.

Mix MC, Farber P, King KI. 1996. *Biology: The Network of Life*, 2nd edition. HarperCollins, New York.

Murray WH. 1988. The application of epidemiology to computer viruses. *Computers and Security*. 7:130-150.

National Research Council (NRC), Committee on Information Systems Trustworthiness. 1999. *Trust in Cyberspace*. National Academy Press, Washington, D. C.

Odum EP. 1953. *Fundamentals of Ecology*. W.B. Saunders, Philadelphia.

O'Neill RV, DeAngelis DL, Waide JB, Allen TFH. 1986. *A Hierarchical Concept of Ecosystems*. Princeton University Press, Princeton, New Jersey.

Overton W.S. 1974. Decomposability: a unifying concept? In: Levin, S.A., ed. *Ecosystem Analysis and Prediction*. Society for Industrial and Applied Mathematics, Philadelphia.

Overton W.S. 1977. A strategy of model construction. In: Hall, C.A.S. and Day, J.W., eds. *Ecosystem Modeling in Theory and Practice:An Introduction with Case Histories*. John Wiley and Sons, New York.

Paine RT. 1966. Food web complexity and species diversity. *Amer. Nat*. 100:65-75.

Peters RH. 1991. *A Critique for Ecology*. Cambridge University Press, Cambridge.

Puccia CJ, LevinsR. 1985. *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Harvard University Press, Cambridge, Massachusetts.

Quirk JP, Rupert R. 1965. Qualitative economics and the stability of equilibrium. *Rev. Econ. Studies*. 32:311-326.

Ricklefs RE. 1990. *Ecology,* 3rd edition. W. H. Freeman and Company, New York.

Shannon CE. 1949. In C.E. Shannon and W.Weaver. *The Mathematical Theory of Communication*. Univ of Illinois Press, Urbana, IL.

Simon HA. 1962. The organization of complex systems. In: Pattee, H.H., ed. *Hierarchy Theory*. Braziler, New York.

Simon HA. 1969. *The Sciences of the Artificial.* MIT Press, Cambridge.

Simon HA. 1973. *The Organization of Complex Systems.* In: Patee, H.H.(ed.) Hierarchy Theory. Braziller, New York.

Summers L., In: Henig, P.D. 2000. Charles Darwin meet Adam Smith. Red Herring. 82:228-232.

Tansley AG. 1935. The use and abuse of vegetational concepts and terms. *Ecology.* 16:204-307.

Ulanowicz RE. 1980. An hypothesis on the development of natural communities. *J Theor Biol.* 57: 355-371.

Ulanowicz RE. 1995. Network growth and development: ascendancy. In: In Patten BC, Jorgensen SE eds. *Complex Ecology: The Part-Whole Relation in Ecosystems*. Prentice Hall, Englewood Cliffs, New Jersey.

Varela FJ. 1979. *Principles of Biological Autonomy.* Elsevier (North Holland), New York.

Walters CJ. 1986. *Adaptive Management of Renewable Resources*. Macmillan, New York.

Walters CJ. and Holling, C.S. 1990. Large scale management experiments and learning by doing. *Ecology*. 71(6): 2060-2068.

Weaver W. 1949. In C.E. Shannon and W.Weaver, *The Mathematical Theory of Communication*. Univ of Illinois Press, Urbana, IL.

Wimsatt WC. 1997. Aggregativity: reductive heuristics for finding emergence. *Philosophy of Science*. 64:372-384.

3 Cyber Ecology Taxonomy

This chapter of the report addresses the most basic level of exploration in which we apply ecological concepts to the classification of malicious code. Higher levels of the community model will be addressed in subsequent sections. In this section, we examine the network from a biological perspective, using insights contributed by biologists to formulate an ecological description of computer networks and attack.

3.1 Why an ecological classification?

The increasing sophistication of cyber attack has outgrown the limitations of the current taxonomy. Dorothy Denning (1999) has suggested that viruses and worms, which she calls *cyberplagues*, "might be better characterized along multiple dimensions according to whether or not they propagate, whether those that propagate do so on their own or require user assistance, and what objects (if any) they attach to (executable files, documents, email messages, boot sectors, and so forth)."

These attributes of computer viruses and worms constitute what biologists would refer to as their natural history. Moreover, the presence of a human user to assist in propagation indicates that there is an interaction among at least two members of a community. In order to assess the effects and implications of the interactive components of cyber attacks, they must be viewed from the perspective of the community, or ecologically. Such models can describe:

- potential breadth of an attack (in terms of lethality, number of species, and species abundance of organisms involved)
- difficulty and points of control
- predictions about future development

Our goal is not to introduce new terminology. The terms *virus* and *worm* are well defined and widely understood. Although the original biological concepts are more complex for living organisms, we do not seek to replace existing paradigms with more biologically accurate analogies. Rather, we will draw on insights into biological organization and hierarchy to suggest ways of understanding and interpreting new, higher levels of complexity in malicious code.

3.1.1 How does an ecological classification differ from other types of classification schemes?

Ecological classifications are based on the relationships that occur within a community. They tend to be more general than other formal classification schemes that attempt to uniquely classify individuals into exclusive categories. A discussion of biological taxonomies and taxonomies of computer attacks is presented as Appendix A. These schemes are often based on physical attributes in biology, or upon mechanistic attributes in computer security. These classifications are informative and necessary, but not complete in their descriptions of effects upon communities and networks.

Ecological communities and computer networks are formed by the interactions of constituents. The interactions vary in strength and may fluctuate over time. As we observed in section 2 of this report, it is often necessary to examine a community from many points of view to acquire as complete a representation as possible of many complex interrelationships. As the lens through which we view the community focuses on specific levels of aggregation, certain relationships may become more or less important and change functionally.

For example, consider the natural community in which lions and gazelles reside. Lions are predators of gazelles. Broadening our view, gazelles are ungulates and consume grass. Widening our view further, humans kill lions. Moreover, humans compete for resources with the entire ecosystem. A description of lions alone does not give us an accurate picture.

3.2 Definitions

Biological organisms and malicious code clearly are not directly comparable. However, their presence and participation in complex webs of effects provides an opportunity to use biological models to generate hypotheses about malicious code. Population and life history parameters are particularly appropriate because they can be used to describe patterns of individual development and relationships among community members. We first provide parallel definitions of terms.

3.2.1 Biological definitions

The Malthusian parameters, *reproduction rate*, *death rate*, and *generation time*, are the fundamental variables used to describe population dynamics. The basic reproduction rate is the number of female offspring produced by one female over her lifetime. Death rate is the number of individuals that die over a specified time interval. Generation time is the time interval between birth and reproduction of offspring.

A *food* (or *trophic*) *web* is a representation of a biological system composed of multiple interacting organisms (a community) through which consumption, but no other relationships can be traced. All the members at one level of the food web feed on organisms at another level of the web. Competition for resources and mutualistic cooperation are not represented.

Communities are formed by the interactions of populations of organisms with each other and their environment. In addition to trophic relationships, mutualistic and competitive relationships are considered.

The *environment* in which an individual exists plays a critical role in its survival. Environmental variables may include weather, soil condition, presence or absence of water, temperature, and available nutrients. Anthropogenic inputs resulting from human activity, such as dams, home construction, roads, and pollutants, are also present in the environment.

Resources are anything at the bottom trophic level of a food web, typically chemical nutrients

Lethality is the loss of reproductive capacity. When an organism dies, its reproductive capacity is destroyed. When an individual sustains a lesser level of loss, corresponding with reduced lethality, its reproductive capacity is impaired.

Intimacy is a loosely defined term that describes the proximity of an organism to its food source. Typically, organisms in a resource poor environment are nonintimate because they must search for many prey. Organisms in a resource-rich environment can enjoy more intimate relationships because their food sources are close at hand.

3.2.2 Cyber domain definitions

Depending upon one's point of view, an individual may be represented as the network, an operating system, an executing or stored program, an email, or other entity. This multiplicity of viewpoints is consistent with ecological analysis, which encourages examination from many points of view to achieve a deep understanding of a system. Individuals may be combined to form systems, and these systems may be combined to form larger systems. We provide four examples of individuals and define birth, death and their environments.

Examples of individuals:

(1) Installed program as an individual

From the perspective of an installed program, an individual is born when the program is installed and dies upon removal. Generation time is the time between installations of the program. The environment includes humans and storage media.

(2) Executing program (process or thread) as an individual

An individual is born when a program begins to execute and dies when execution stops. Organisms can survive in a quiescent state, where they are present on disk as source or object code, but not executing. The biological parallel of such a quiescent state may be a rhizome or plant bulb. Generation time is the time between executions. The environment includes humans, storage and powered-on computers.

(3) Static code as DNA; computer as a cell

The metaphor can be extended to a microbiological level, where the compiled or interpreted code represents DNA. Interpreters analyze and interpret code as ribosomes interpret DNA for the manufacture of proteins in biological organisms. The computer is parallel in structure to a cell. A network of computers constitutes an individual. The network is born when the cells combine and interact and dies when the network ceases to function in a coherent manner. The environment includes humans, storage, powered-on computers, and network connectivity.

(4) *Any actor represented as an individual*

Examples of actors include CPU, programs, host computers, networks, and humans. Definitions for birth, death, and the environment are dependent upon the specific models.

Resources include CPU cycles, bandwidth, and memory. All of these are provided by computers and consumed by programs.

Food webs (also called *trophic webs*) are expressed in terms of resource consumption. In computer networks, resources such as bandwidth, CPU cycles, and memory are consumed by other programs.

Lethality is measured in terms of damage inflicted that reduces functionality. It applies to a time interval, since computers and code may be replaced or repaired.

Intimacy is a measure of remoteness. For example, code may be executed remotely (nonintimate contact) or locally (intimate contact).

3.3 Construction of the ecologically-based classification

In this section, we demonstrate how a classification of malicious code-based information might be constructed using Malthusian-like parameters, as well as information about damage caused by malicious code and mechanisms employed. We begin by defining features of malicious code (variables). We collected data using publicly available descriptions of computer viruses and worms found on the World Wide Web, mostly published by virus scan companies. The classification was constructed automatically from data using machine learning techniques.

3.3.1 Variables

In order to classify malicious code, we defined 16 variables based on the perspective of an installed program as an individual (the first example of an individual discussed in the preceding section).

The biological parameter, lethality, was addressed by questions concerning loss. Lethality is expressed as damage to programs and can be inferred from the following questions:

1. What is the potential damage to data and/or hardware?
2. What is the typical loss of time (availability and recovery)?
3. What is the potential for loss of confidentiality?

Damage to data and/or hardware was classified as nuisance, deletion of relatively unimportant data files, deletion of system files, or deletion of entire file systems or nonfile data such as partition tables and boot sector. Time lost was evaluated with respect to the estimated time necessary for recovery, in minutes, hours, days, or weeks. The presence of a confidentiality attack, that is, the possibility of unauthorized information transmission to another program, was noted. Information about reproduction was captured by questions about replication and spread:

4. Does it replicate?
5. How does it spread?
5a. Does it spread actively?
5b. Does it arrive via an abnormal transaction?

Malicious code was said to replicate if copies of the code were produced, for example using MS Outlook or IRC. Spread was described as active when the code was dispersed by the program itself, through email or some other transaction. Spread was inactive, or passive, if the program was incapable of supporting its own dispersal and

relied on factors in its environment, such as humans using a computer floppy disk to copy files from one machine to another. For programs that spread actively, the manner in which the program transmitted itself was also noted. Transmission through a corrupted or unauthorized transaction was labeled abnormal.

The life cycle of the malicious code was described by responses to the following questions:

6. Where does it reside?
7. How is it run? (What causes execution?)
8. Does it persist?
9. How is it activated?

The location of the infecting program was described as one or more of the following:

- Runs as a new executable program (.exe, scripts, .com files)
- Runs as an old executable program (changes .dll files, autoexec.bat)
- Runs when user accesses data (macro)
- Reconfigures (.ini, changes registry files)
- Nonfile (partition table, boot sector, bios (PROM, flash), hvram)

These are shown graphically in Figure 5.

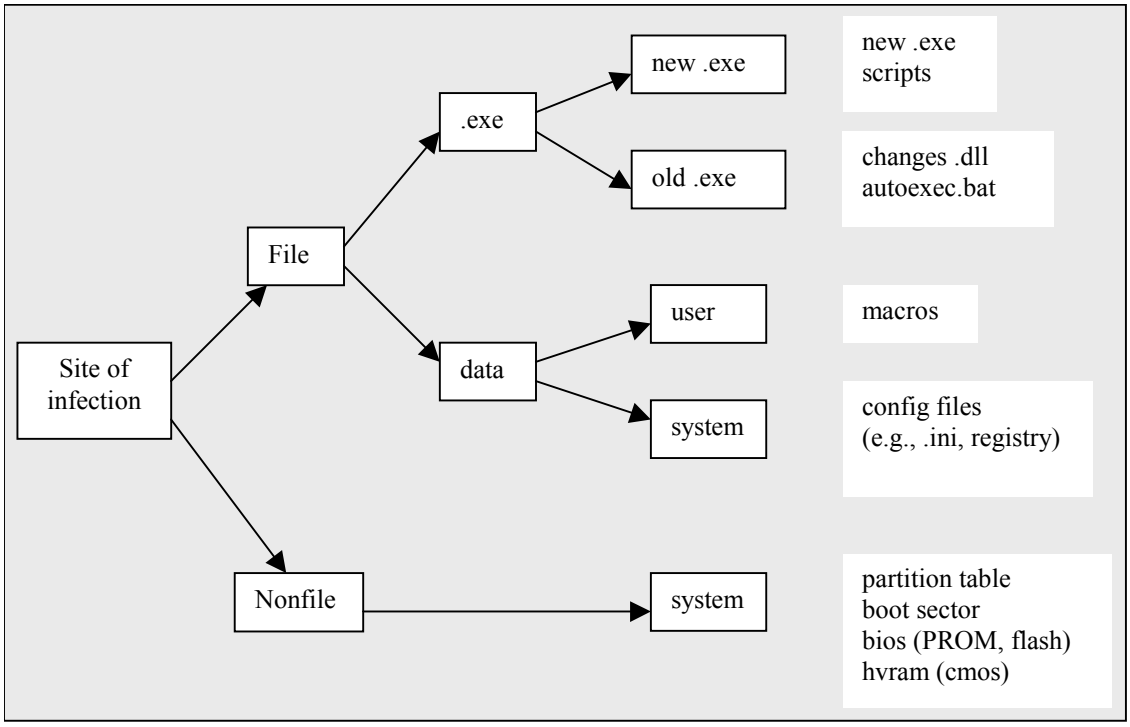


Figure 5. Classification of sites of infection by malicious code

The cause of execution was observed to be the user, autonomic (e.g., scanned network shares), or a transmission event (e.g., email). Persistence, meaning that removal was necessary, and whether the code was permanent or transient was noted. The requirement for human intervention, for example by opening an email attachment, was recorded, as was information about the environment:

10. What is the OS?
11. When was it discovered?

The population dynamics of infecting programs and hosts over time were captured by the following questions:

12. Is it currently extant?
13. How quickly did it spread upon initial release?
14. How wild was it initially?
15. How wild is it now?

Wildness refers to the extent to which malicious code is spreading among computer users. It incorporates the number of sites and computers, and geographic distribution of these sites.

To infer fitness, we collected information about the success of the malicious code in evading detection.

16. How likely is it that users will notice:
    An associated event upon receipt? (e.g., a suspicious email)
    An event associated with infection? (e.g., a message or display)
    Functional problems?

3.3.2 Data

We gathered data for 24 computer viruses and worms, mostly using sources found on the World Wide Web (www.symantec.com, www.mcafee.com, www.f-secure.com, and others). The viruses and worms for which data were abstracted were: Navidad, myna, Michelangelo, Morris worm, Macmag, Kakworm, Funlove.4099, Happy99, Scary, Pretty Park, VBS.Network, VBS.Loveletter, VBS.Stages, MTX, Qaz, Sonic worm, CIH, Marlburg, M97M.Chack, W97M.Onex, StrangeBrew, RemExp, HLLC.plane, and Christmas.

3.3.3 Classification

Using these data, we constructed a classification scheme automatically using machine learning techniques. Compared to the manual construction of a classification, the automatic construction does not require human expertise and tends to be less *ad hoc*. It usually requires an abundance of data, and tuning the parameters of machine learning algorithms is often a difficult art. In the current work, we constructed a preliminary classification.

Our construction of the classification consisted of two parts. First, we used a clustering algorithm to find a specific number of clusters (classes) of malicious code.

Then, we constructed a decision tree so that we could determine a class of a malicious code based on its features. Both processes were data-driven.

3.3.4 Clustering

The goal of the clustering algorithm was to find a specific number of center points, each of which represents a cluster or a class, using data points so that the sum of the squared distance from each data point to its nearest center point is minimized. As we saw earlier, data are recorded as vectors of variables or features. Let *d* be the dimension, that is, the length of a data vector, and *m* be the number of data points. Then, minimizing the sum of squared distances is the same as minimizing the normalized squared distance *E* defined as follows:

*E = sqrt((sum_i=1 to m square_distance(nearest(ith_point), ith_point))/(m*d)),*

where the squared distance of two points is defined as follows:

square_distance(c, p) = sum_i=1 to d square(c(i)-p(i))

For the clustering algorithm, we used an unsupervised learning technique, the maximum-neuron-based (Takefuji *et al.*, 1992) self-organization classification algorithm (Oka *et al.*, 1996). This algorithm converges faster than the more conventional Kohonen's self-organization map (Kohonen 1993).

For each number of centers, we chose the best of 10 trials. The following table shows the result of this experiment. The first column shows the number of centers. The second column shows the error *E* defined above. The variable class represents the resulting class assignment for each data point. The third column shows the entropy of the class variable. Entropy is the amount of information contained in that particular class variable. The fourth column shows the feature with the largest mutual information with the class variable, and the last column shows the feature with the smallest mutual information with the class variable. Mutual information indicates the degree to which knowing about a feature variable informs us about a class variable. They represent the most and least relevant features with respect to the class variable, respectively.

Table 4. Results of the clustering algorithm

Number of centers	Error	H(class)	Variable = largest mutual information	Variable = smallest mutual information
1	0.357	0	0	rep, whereother=0 for all cases
2	0.343	0.98	active=0.344	
3	0.323	1.53	rcptevent=0.395	
4	0.31	1.9	extant=0.391	
5	0.303	2.04	os=0.658	
6	0.292	2.49	os=0.726	
7	0.279	2.76	os=0.679	
8	0.269	2.64	dam=1	
9	0.261	3.02	os=0.825	
10	0.245	3.26	dam=1.09	
11	0.229	3.27	dam=1.02	
12	0.229	3.47	os=1.07	
13	0.21	3.64	os=1.29	
14	0.204	3.55	dam=1.17	
15	0.21	3.67	os=1.07	
16	0.198	3.86	dam=1.24	
17	0.185	3.77	dam=1.35	
18	0.171	3.97	os=1.24	
19	0.155	4.08	dam=1.55	
20	0.155	4.08	dam=1.55	
21	0.153	4.08	time=1.45	
22	0.108	4.3	time=1.5	
23	0.125	4.14	time=1.45	
24	0.13	4.33	dam=1.55	

Key to variable names: active = Does it spread actively?; rcptevent = Is it likely that a user will notice an associated event upon receipt?; extant = Is it extant?; os = What is the operating system?; dam = What is the potential damage to data and/or hardware?; time = What is the typical loss of time?; rep = Does it replicate?; whereother = Where does it reside? (other)

Formally, the entropy, conditional entropy, and mutual information are defined as follows:

entropy: H(class) = sum_i=1 to N -p(class=i)log2 p(class=i)
conditional entropy: H(class|feature) =
sum_j=1 to |dom(feature)| p(feature=j)
[sum_i=1 to N -p(class=i|feature=j)log2 p(class=i|feature=j)]
mutual information: I(X;Y) = H(X) - H(X|Y)

The following graph (Figure 6) shows the plot of *E* and *H* in the above table. The error *E* is an approximate linearly decreasing function of the number of centers for this data set. That implies that there is no "natural number" of clusters in this data set. If the

number of natural clusters were three, for example, the graph would be flat after three. The graph for the entropy *H* indicates the resulting class distribution is consistently close to uniform, which is often preferable.

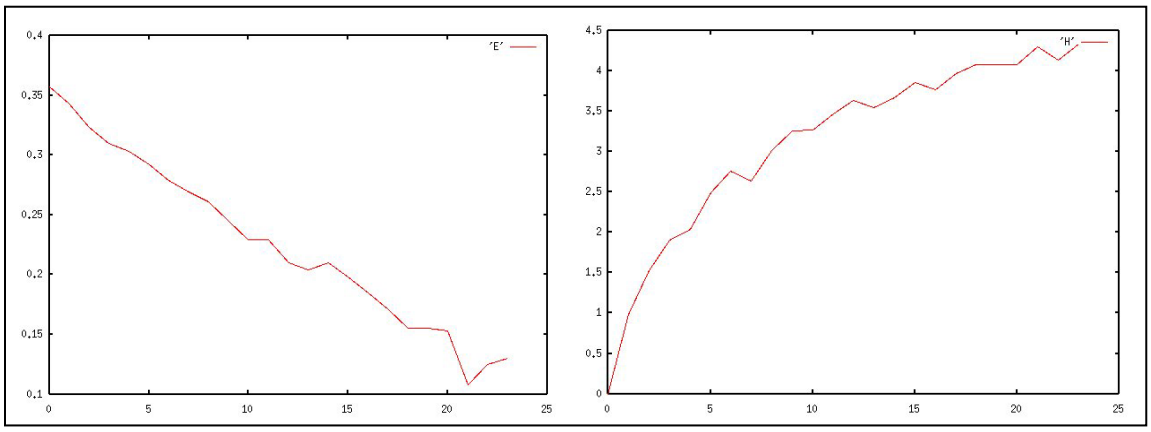


Figure 6: Graphs of *E* (left) and *H* (right)

According to the above table, the feature with the largest mutual information varies, but some patterns can be seen. The three features, damage, operating system, and time, consistently have large values of mutual information. Two features, rep and whereother, do not possess significant mutual information with class, meaning that they can be safely ignored for the purpose of classification.

3.3.5 Decision Tree

We created a decision tree in which intermediate nodes represented features and leaves represented classes. In order to construct a concise and generalizable decision tree, we used mutual information. At each level, as the branching node, we chose the feature with the largest mutual information for the class variable with respect to the data belonging to the corresponding subtree.

The following decision tree was created for the classification with three classes. In the tree, each intermediate node is labeled by a feature, and each arc is labeled by a value. The leaves contain the class number with examples of malicious code possessing all the attributes specified along the path, from the root of the tree to the specific leaf. For example, if it is likely that users will notice an associated event upon its receipt (rcpevent=likely), and it spreads slowly (spread=slow), the malicious code is classified into Class0, which contains the "Scary" virus.

The decision tree created used only 4 out of 16 available features (Figure 7). It was compact and easy to inspect. It was also generalizable and readily applied to new data.

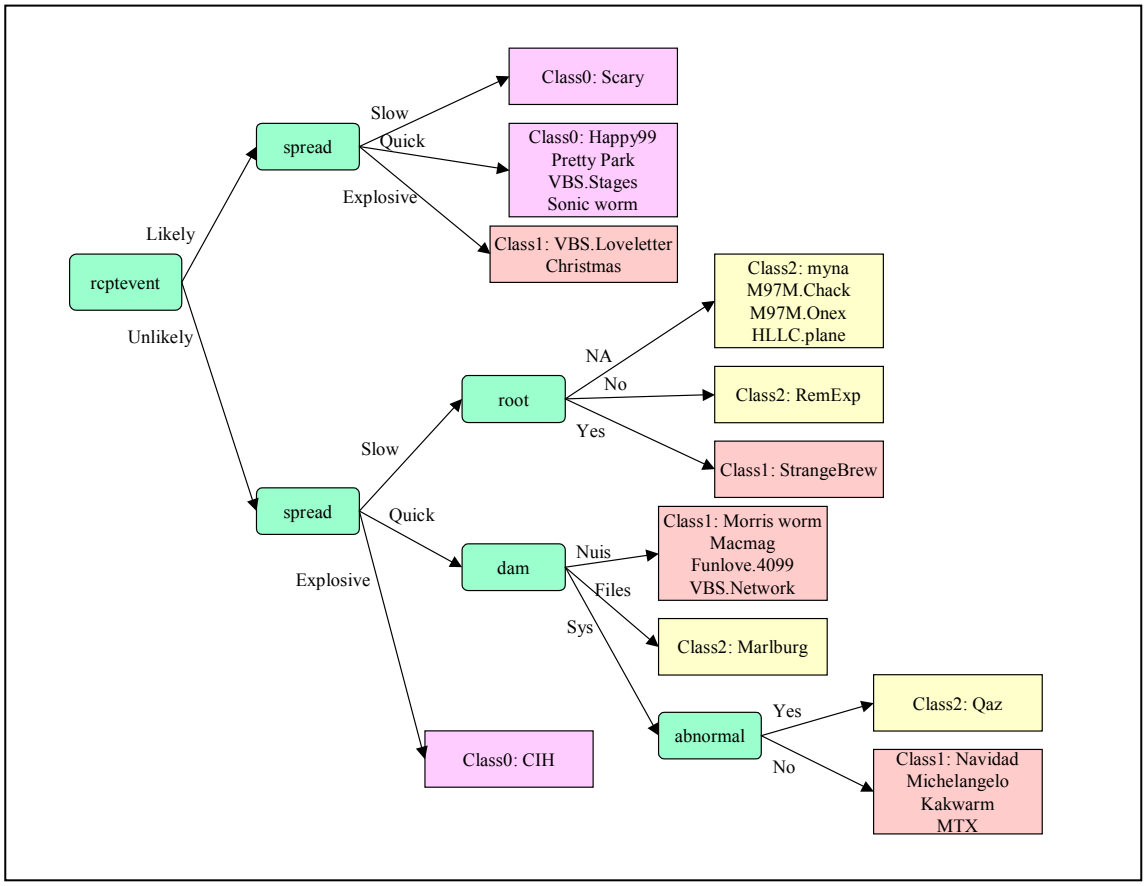


Figure 7. Decision tree with three classes

In this section, we demonstrated the construction of a classification of malicious code based on ecological features using machine-learning techniques. Since we were dependent upon well-described cases, our data were not representative of the entire body of past and existing malicious code. In order to obtain more interesting results, more data are necessary. Future research into this method of classification will require creation of a larger data set and tests of the usefulness of the results, such as prediction of unknown feature values using known feature values.

3.4 Application of ecological theory to cyber ecological classification

In this section, we discuss the general classification of malicious code using ecological principles.

3.4.1 Biological classification

Systematics is a biological discipline whose major goals are to describe biological diversity and to produce natural classifications based on relatedness (Marcus 1993). These goals are accomplished through the discovery and identification of living and fossil

organisms. Taxonomy is the theory and practice of classifying this biological diversity (Chernoff 1986).

Biologists construct taxonomies based on various types of information, including physical characteristics observed during examination of living and preserved specimens, observations about behavior and development, and information about habitat. Artifacts, such as nests or ectoparasites, may also be used in a scheme. Any aspect, although mostly inherited, can be observed and compared among organisms.

For systematists, the most basic unit of classification is species, although this unit is highly controversial. This simple concept is at the center of intense controversy among systematic biologists. The three prevailing definitions are biological, phenetic and evolutionary (or phylogenetic) species. *Biological* species, originating from Ernst Maier, are distinguished by their reproductive isolation. Members of any one biological species can exchange genes within that species, but not with other species. *Phenetic* species are described by their morphology and location. Within a given geographical area, members of a phenetic species will share morphological characteristics that are distinct from other populations. Evolutionary species are defined in terms of their history. Members of a *phylogenetic* species share a history that can be differentiated from the history of other species. Species is typically the finest grain of a hierarchical classification scheme. In ascending order, the classification extends to genus, family, order, class, phylum and kingdom.

A taxon is any formal unit in the taxonomic system. The classification of taxa is accomplished through the shared features, or characters, of organisms and lower-level taxa. Characters may be nominal in nature (such as shape, color, or pattern), qualitative (such as long or short), or counts of discrete features (such as the number of toes). The presence or absence of a feature, such as an amino acid in a protein, is also used.

Biological and computer taxonomies are very different for a variety of reasons:

(1) Computer viruses have no equivalent of DNA. They consist of code in which it is easy to recognize uniqueness, but difficult to assign membership to larger groups;
(2) Malicious code lends itself easily to descriptions of mechanism. Biological organisms, because of evolution, exhibit structural and genetic characteristics that can be used for classification;
(3) History. Biological classification can be traced back through thousands of years. The classification of computer agents is strictly a modern endeavor.

3.4.2 Classification based on trophic strategy

For ecologists, the identification of taxa assumes a secondary role to the identification of an organism's placement within the ecosystem. The location of the organism in the food chain (i.e., its trophic relationship with other members of the ecosystem) and its functional role within the community are of paramount importance. The functional classification of organisms includes:

- Producers – mainly photosynthetic organisms responsible for the community's net primary productivity.
- Herbivores – primary consumers that feed directly on photosynthetic organisms.
- Carnivores – prey on herbivores (as well as other carnivores).
- Scavengers – feed on organic refuse or carrion.

- Parasites and parasitoids– consume the tissues of living hosts. Parasites feed on their hosts, but are not a lethal burden to them. Parasitoid adults lay their eggs in host organisms. The young feed on the living host until they reach adulthood. When the adults emerge, the host is killed.
- Decomposers – fungi and bacteria that break down organic debris.

An alternative ecological classification scheme uses life history parameters (i.e., birth rate, death rate, time between generations). We present a brief explanation of the mathematical derivation of the ecological classification. For a more detailed analysis, please refer to *Trophic Evolutionary Pathways: A Model Based on Life History Parameters* (Morris and Rossignol, submitted). The most basic parameters for describing population dynamics are the three Malthusian parameters: survival (*l*), fecundity (*m*), and generation time (*t*). They can be used to model population growth with unlimited resources in the Euler, or Lotka-Euler equation (Wilson and Bossert 1971):

$$1 = \sum_{t=1}^{\infty} l_t m_t e^{-rt}$$

where l_t represents survivorship, m_t is fecundity, t denotes generation, and r is the intrinsic rate of population growth. The Malthusian parameters may be observed empirically and used to determine the intrinsic rate of population growth, r.

The intrinsic rate of growth, r, describes the potential growth of a population without consideration of density dependence. It is the per capita difference between births and deaths over a fixed period of time. The fitness of an organism is determined by its intrinsic rate of growth. In the absence of constraints, an organism need only increase its fecundity to attain a higher rate of intrinsic growth and increased fitness. However, in the face of limited resources, organisms must make tradeoffs to maximize fecundity while reserving enough energy to stay alive. These tradeoffs have been represented in the controversial *r*- and *K*- model of selection. Organisms that evolve a high rate of fecundity at the expense of longevity are said to be *r-selected*. These organisms have many young, but short life spans. Those organisms that have evolved a lower rate of fecundity while being able to breed for multiple seasons are termed *K-selected*. These organisms have few young, but longer life spans. The *r-K* model has been problematic, and does not seem to apply to parasites that exhibit both *r*- and *K*-traits. They invest reproductive energy into laying a large number of eggs (an *r*-selected trait), yet also into increasing egg size (a *K*-selected trait). Many adult parasites are long-lived (another *K*-selected trait).

Calow (1983) reconciled this discrepancy by demonstrating that parasites adhere to the *r-K* selection model, but within the context of a relatively rich nutrient environment. That is, they are not faced with the same nutrient-limiting constraints as other free-living organisms because they essentially live in their food. So, an *r*-selected parasite that lays many eggs may be relatively long-lived because of the abundance of available nutrients.

For the ecological classification, organisms are described according their trophic strategies, that is, the methods in which they feed. Trophic strategies are grouped into four general categories, each corresponding to a mode of feeding. Predators consume

multiple prey items throughout their lifetimes. Grazers consume parts of many organisms in a non-lethal way. Parasites also consume nutrients derived from their hosts in a non-lethal way, but are in more intimate contact with a single host. Parasitoids derive nutrients from one host while in intimate contact, but eventually kill them.

Pairs of organisms can be described along the dimensions of lethality and intimacy (Figure 8). The gazelle-lion, prey-predator relationship is classified as lethal and nonintimate. The gazelle is killed by a nonintimate organism, one with whom it does not have close contact on a protracted basis.

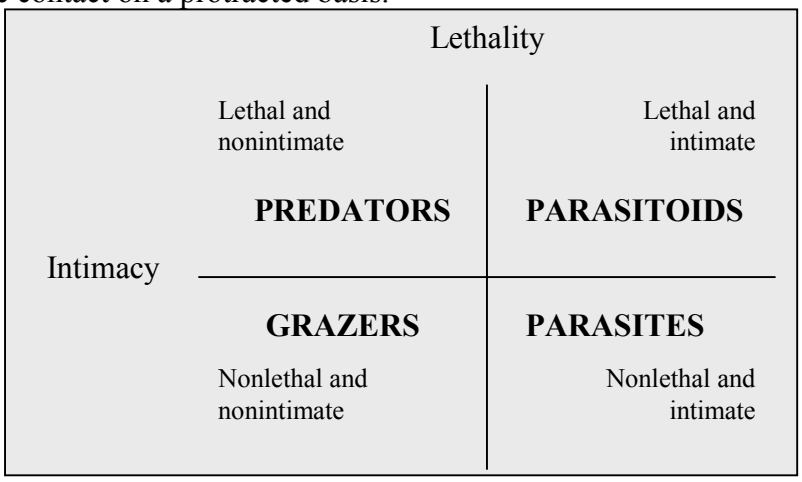


Figure 8. Ecological classification of organisms

The ecological classification allows for individual organisms to appear more than once in the classification scheme as members of different producer-consumer (e.g., grass-gazelle) or prey-predator pairs. The strength of this classification is that it represents the relative position of organisms in particular contexts within the structure of the ecosystem.

3.4.3 How does this classification apply to computer networks?

At face value, the biological analogy does seem to apply. Cyber attack agents do seem to vary with respect to the damage they inflict upon target computers (lethality) and site of execution (remote or local; intimacy). Agents such as viruses transmitted in email attachments seem to be *r*-strategists, while dedicated confidentiality attacks, which are fewer in number and more difficult to launch, seem to exhibit characteristics of *K*-strategists.

The analogy does not transfer completely, however. All biological organisms reproduce. Their trophic strategies reflect the manner in which they obtain energy for survival and reproduction. The same cannot be said for cyber attack agents, some of which do not derive nutrients for these purposes from the attack. By design, they may or may not be capable of replication and this figures prominently in the breadth of attack.

In order to capture the biological nature of cyber attack among agents that reproduce and to summarize the trophic-like strategies of those that do not, we stratified the ecological model into two layers: code that replicates, and code that does not. These are summarized in Table 5.

Replicating code exists that satisfies the criteria for cyber parasitoids, cyber parasites, and cyber grazers. We have not found an example of replicating code that

fulfills the criteria for a predator. The remote use of host resources for replication is problematic. How can the resources of a host on which code does not run be used to spread that code? Code generally uses the resources of a host by running on that host. The only case we have found of a remote, nonintimate relationship where code can use the resources of a host to replicate and spread without executing on that host is through the use of open network shares. The code uses the interrelationship between client and server to inject itself into the network.

Nonreplicating code fulfills the criteria for what we term *pseudo-trophic* relationships. We use the term *pseudo-trophic* to describe relationships where the entity of interest does not replicate. Here, examples of predation do exist. For example, damaging, remote denial of service attacks are examples of pseudo-predators since they act as predators, but do not replicate.

Table 5. Classification of replicating and nonreplicating malicious code using ecological classification

	Replicating		Nonreplicating	
	Intimate	Nonintimate	Intimate	Nonintimate
Lethal	Parasitoid	Predator	Pseudo-parasitoid	Pseudo-predator
Nonlethal	Parasite	Grazer	Pseudo-parasite	Pseudo-grazer

We describe code in each category of the classification in Table 6. The dimensions of lethality and intimacy, which were defined earlier in the previous section, lie on a spectrum spanning values from low to high. At the extremes, they are clearly distinguishable. Moving away from the extremes, however, differences become less distinct. We have attempted to provide examples that are clearly separated as possible.

The flexibility inherent in the description of a single entity in multiple categories allows for appropriate description from different perspectives. For example, in Table 3 above, from the perspective of the network server, a virus that spreads through open network shares is a cyber grazer. It is consuming resources remotely in a nonlethal manner. However, from the perspective of an individual computer served by the server, the same virus may be a cyber parasite. This ability to view relationships at multiple levels is consistent with the systems-level perspectives required to represent the network. As we incorporate broader areas of the network into our models, the relationships change. This multilevel approach also seems applicable in terms of network-centric warfare. Alberts, Garstka and Stein (1999) contrast platform-centric and network-centric warfare. In platform-centric warfare, platforms "own" weapons and weapons own sensors. In network-centric operations, platforms, weapons and sensors can be reconfigured dynamically to achieve a commander's intent. Decision makers and actors can assume different roles depending upon the roles they assume in fast-paced battlespace domains.

The basic ecological model of malicious code is not entirely foreign. Adleman (1988) constructed an ecological taxonomy of viruses using the dimensions of pathogenicity (producing injury) and contagiousness (ability to spread). He classified viruses into four disjoint, and therefore independent and mutually exclusive, categories: benign, Epeian (after the builder of the original Trojan horse of *The Odyssey*), disseminating and malicious. Adleman's taxonomy contained the germ of an ecological

taxonomy because it took into account the effect one type of virus had upon another. For a more detailed discussion, please refer to the appendices.

Table 6. Cyber classification based on trophic strategies

Description	Classification and examples
Intimate, lethal agents, cyber parasitoids, exist in close proximity with their host computers and inflict severe damage. Like their biological counterparts, they may allow the host to live until it is time to execute their payload.	Cyber parasitoid (replicating) e.g., virus that inflicts severe damage
	Pseudo-parasitoid (nonreplicating) e.g., Trojan horse, logic bomb
Intimate, nonlethal agents, cyber parasites, exist in close proximity with their host computers, but do not cause severe damage. Nonreplicating cyber parasites are merely annoying. We include them even though they are not malicious.	Cyber parasite (replicating) e.g., most viruses and worms
	Pseudo-parasite (nonreplicating) e.g., Javascript in web pages (e.g., pop-up ads)
Nonintimate, lethal agents were discussed above. They use host resources remotely to replicate.	Cyber predator (replicating) – no examples
	Pseudo-predator (nonreplicating) e.g., distributed, remote denial of service attack
Nonintimate, nonlethal agents "harvest" resources from many computers, but do not inflict severe damage.	Cyber grazer (replicating) e.g., virus that spreads by open network shares on server[1]
	Pseudo-grazer (nonreplicating) e.g., unsolicited commercial email (spam)

3.4.4 Cyber parasites

During the course of our investigation of an ecologically based taxonomy of malicious code, we examined closely the characteristics of computer viruses and worms. Many of these agents fall into the quadrant corresponding with parasites in the ecological taxonomy. We have termed these nonlethal, yet injurious, cyber agents *cyber parasites*.

We define a cyber parasite as computer code that:

(1) is intimate with its host (that is, its code constitutes a locus of control that executes on the host), and
(2) replicates without the necessity of human intention or awareness.

[1] *From the perspective of an individual machine, such an attack might be intimate and damaging (parasitic).

To date, most cyber parasites have been considered harmful by most people. In the event that beneficial agents satisfying the two criteria listed appear in the future, the following third characteristic should apply:
(3) inflicts damage.
We discuss beneficial cyber parasites in Section 3.5.4.

The concepts of trophism (feeding) and lethality must be generalized to span both the biological and cyber domains. In biology, trophism (how an organism feeds) is a method of energy transfer. By feeding on an organism, energy that was not available previously is liberated for use by the consuming organism. Cyber parasites consume resources such as bandwidth, CPU cycles, and memory from the hosts they infect. As in biological parasites, this consumption of the host's resources is not lethal.

3.4.5 Implications of cyber parasitism

The model of cyber parasitism allows us to examine malicious code from an ecological perspective. We briefly discussed the nature of parasitism in section 1 and will elaborate on the characteristics of parasites in general in this section.

The life cycles of parasites are often complex. Although they spend much of their life spans in intimate contact with their hosts, the young must find new hosts to infect. Many parasites infect more than one organism in their lifetimes. They are vulnerable in this dispersal phase as they travel between hosts.

Parasites and their hosts have evolved together over time, allowing parasites the opportunity to find a myriad of pathways through which to infect. Again, we emphasize that the mere presence of actors in a community can be noninformative. The nature and scope of infection is determined by the structure of the community and the manner in which actors interact within this structure. The long association between biological parasites and hosts has allowed the development of a rich structure within which many different transmission paths have evolved. These paths range in complexity from simple to contorted. The more complicated transmission paths involve entire communities of organisms. To our knowledge, the community model for disease transmission presented here is novel to both ecology and cyber ecology.

3.4.6 Vector transmission

In the transmission of vector-borne parasitic disease such as malaria, a vector, such as a mosquito, transmits the disease to a human. The mosquito itself is also a pest that feeds on humans. Malaria exploits the relationship between human and mosquito to propagate. The malarial parasite uses a grazer, in a nonintimate, nonlethal relationship with humans, to bring it into intimate contact with its final host.

The equivalent transmission of malicious code occurs when it is inserted into a legitimate communication, such as email, and transmitted to another computer. As in vector-borne diseases such as malaria, the malicious code exploits a relationship between human and email. In nature, organisms are either vectors or they are not. Among cyber agents, varying degrees of vector-like behavior can be observed. At the simplest level, the infection is brought into direct contact with the host. Examples are the transport of a virus on a floppy diskette or as an email attachment. Widespread use of mobile code will create opportunities for clearer and more literal vector transmission (i.e., a cyber parasite might infect a segment of mobile code in order to reach a new host).

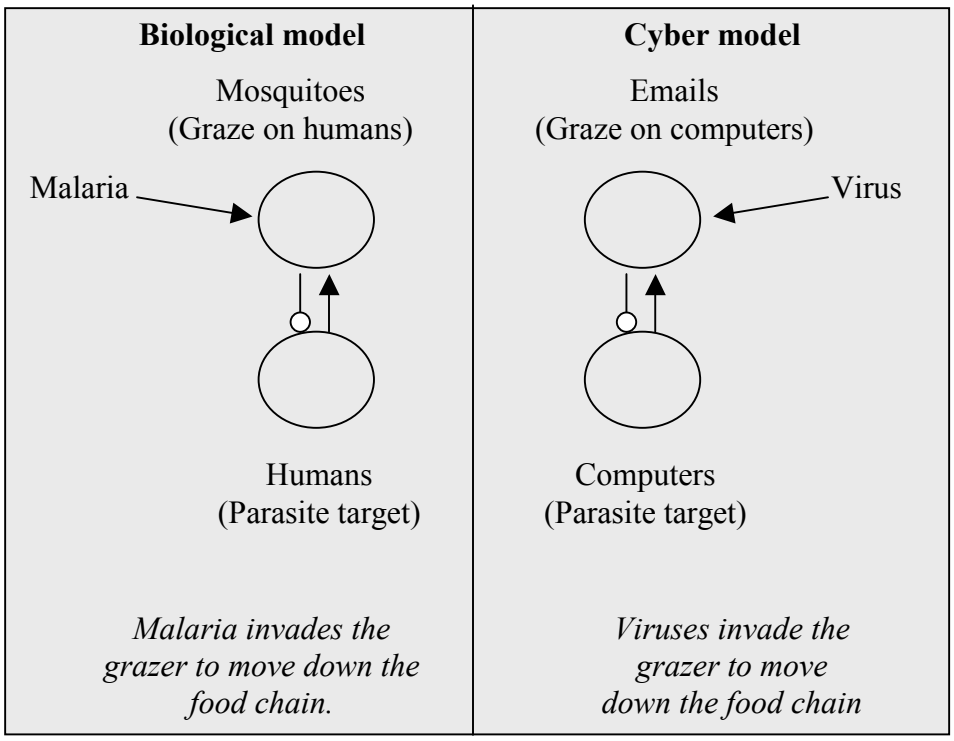


Figure 9. Biological and computer vectors of disease

Vector-borne transmission is depicted graphically in Figure 9 as signed digraphs. Positive effects, those that cause an increase to one species from another, are represented with arrows. Negative effects, those that cause a decrease to a species are represented with lines terminated in circles. An arrow indicates that one variable causes another to increase. A line terminated in a circle indicates a decrease from one variable to another. In this model, not all mosquitoes are infective, and it may take multiple bites to transmit the infection. In the cyber model, we likewise assume that not all emails are infective. Emails are grazers because each one requires the use of a computer's resources (memory, CPU cycles), but do not cause the machine to cease functioning.

3.4.7 Complex community transmission

Complex biological communities may also include parasites. In these systems, the parasite depends on a number of sequential events to survive. These organisms have evolved in tandem over long periods of time, providing evidence for the presence of stability that serves as a backdrop for this coevolution. With increasing community complexity, simple low probability steps in transmission from intermediate to final host are replaced with longer chains of higher probability ones. The relationships have developed because historically there has been a significant likelihood that the more complex chain will be maintained.

Dicrocoelium dendriticum, the lancet worm, infects through a complicated transmission cycle. The parasite is a trematode, which must live in snails during part of its life cycle. To reach its ultimate host, a sheep, it must negotiate a tortuous cycle that begins when it is deposited in grass as the snail defecates. The snail feces are then eaten

by ants. In the ant's gut, the parasite divides into four sister clones. One of the clones makes its way to the ant's brain, where it modifies the ant's behavior, causing it to crawl to the top of a blade of grass and cling to it, behavior that the ant would normally exhibit during a storm. At the top of the blade of grass, the ant is more likely to be eaten by a grazing sheep. The parasite is eaten and reproduces in the sheep, depositing its eggs in feces that are eaten by snails. *Dicrocoelium dendriticum* has found an indirect way to enter the grazing relationship between the sheep and grass. The community for this transmission cycle is shown in Figure 10.

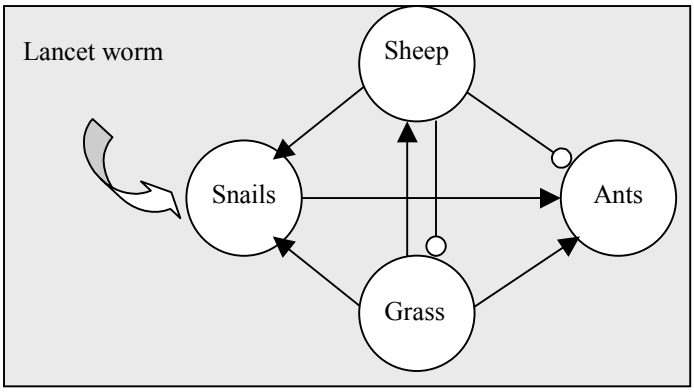


Figure 10. Model of a community-level parasitic relationship

The mechanistic approach now taken to describe computer viruses and worms does not lend itself to recognition of these more complex transmission communities. In Figure 11, we present a possible model of a community that is vulnerable to community level parasitism.

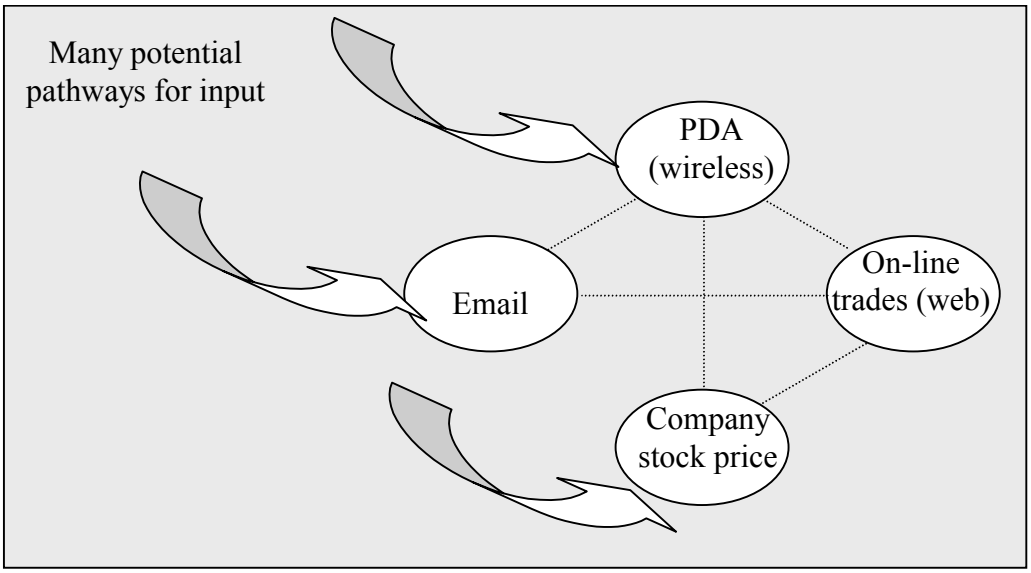


Figure 11: Signed digraph of a network vulnerable to community-level parasitism

Figure 11 sketches the inter-relationships among emails, wireless transmission, on-line trading and company stock price. This network is vulnerable to community-level parasitism because cyber parasites may breach the system at any node and insert themselves into legitimate transaction streams. In this way, cyber parasites may impact their desired targets indirectly.

Indirect transmission paths have been described as *indirect coupling,* in which a virus is injected into the most accessible unprotected point and spreads to its objective target. For example, a virus could be injected into a tactical data link, where it is transmitted to a tactical aircraft and propagates to a Command and Control Facility. This is similar to the community-level model in that a parasite can be transmitted along legitimate transaction paths. Community models, however, also incorporate indirect effects that depend on changes in behavior as well as propagation through links. *Dicrocoelium dendriticum*, the lancet worm, for example, modifies the behavior of an ant so that it is likely to be eaten by a sheep. In a community model, chains of effects, as well as chains of transmission, are important.

An interesting trend that we have recently observed is a composite attack consisting of modular malicious code such as MTX, consisting of a virus, a worm, and a backdoor. Perhaps written by multiple authors, this type of malicious code represents a new level of sophistication. The modules are specialized and work together in a focused attack. The viral code infects, the worm code replicates, and the backdoor is the payload. While not a community *per se*, such new compound agents may represent movement toward more coordinated malicious code. Given the possible diffuse nature of such a distributed attack, it may be difficult to assess the potential effects. Observation of system-level response will be necessary to detect such attacks in time to minimize damage.

3.4.8 Are cyber parasites protective?

A controversial argument has been made in biology that parasites protect organisms from autoimmune disorders. The reasoning is that parasites and their hosts have struck a delicate balance. When the parasite is removed and the host's immune system is no longer challenged by the parasite, the host will attack itself. In this argument, the human immune system is tuned to behave optimally (for the human) in the presence of parasites. This has happened because over evolutionary time scales, the presence of parasites has been normal for humans. This is not the case for computers and computer networks, which are still designed assuming the complete absence of parasites.

Cyber parasites may confer a competitive advantage to some members of the network. Experimental evidence has shown that organisms that allocate significant resources to resistance against parasites may be less successful than those without defenses who accept the damage and allocate resources to compensate. Resistance can be so expensive that it is not worth maintenance and it may be cheaper to yield losses. There is a need for balance.

Cyber parasites may indirectly mitigate loss by encouraging vigilance. The fact that they are common protects us to some extent. Seeing malicious code all the time prevents us from seeing it the first time at a particularly bad time. Michael O'Hanlon (in Schwartz 2000), a senior fellow and the Brookings Institution noted, "People do you the

favor of attacking you so often that you have a chance to build up protection. . . . It's a great way to figure out where your weaknesses are."

3.4.9 Have we observed evolution among cyber parasites?

We have not observed coevolution between cyber parasites and their hosts nor do we have evidence that cyber parasites have driven the evolution of their hosts. Cyber parasites to date have been targeted to specific hosts. Although aspects of the host change over time, for example operating systems are continually updated (Figure 12), cyber parasites have not tracked these changes across successive versions. They have been individually designed to follow applications that run under updated operating systems. Evolution through natural selection cannot be targeted. It is the result of many random mutations, most of which are unsuccessful.

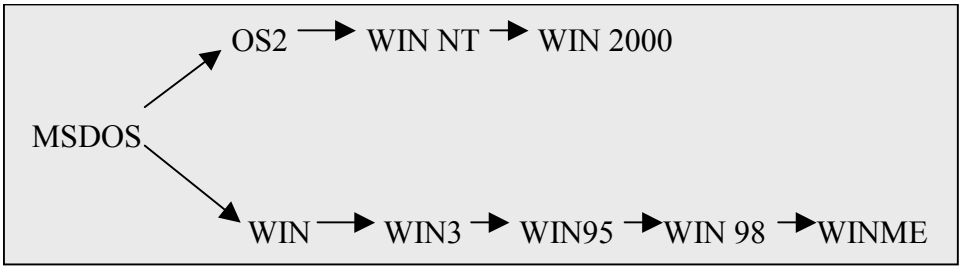


Figure 12. A family tree of operating systems

3.5 Future trends

3.5.1 Community attacks

Wherever there is a longstanding, legitimate interaction or proximity between two members of a community, there exists an opportunity for parasitism. In ecological communities, parasites and their hosts have coevolved over long periods of time. In computer networks, evolution, in the sense of change over time, has been driven by human coders. As the level of sophistication increases, the breadth of communities affected by cyber attack will surely grow. Anticipating all possible mechanisms of infection is intractable. Indeed, response to new cyber agents tends to be reactive, with a continuous parade of novel and imaginative malicious codes. We believe that the visualization of community structure and the associated pathways for infection will yield a more tractable solution space to the problem of diagnosis and treatment of cyber parasitic infections as well.

During the week of 18 December 2000, news reports about the behavior of Kriz were published[2]. A polymorphic virus spread through what we term direct transmission, Kriz was reported to have increased its range by attaching itself to worms, such as the Happy99.worm and Bymer.worm, and using these to spread. If this is indeed the case,

[2] 'Kriz' virus waiting for Christmas strike.
http://www.cnn.com/2000/TECH/computing/12/21/kris.virus.idg/index.html
Kriz virus makes a return appearance. http://www.zdnet.com/zdnn/stories/news/0,4586,2666836,00.html

then we will have observed an instance of community transmission occurring autonomously in the wild.

More recently, a variant of the Klez worm, Klez.h, combined with older viruses to form 'Klez cocktails'. W95.CIH.1049, a slight variation of the Chernobyl virus, has been detected in recent infections of the Klez worm. Vincent Weafer, senior director of Symantec's Security Response Team noted, "As far as [Chernobyl] is concerned, the Klez worm is just another file to infect. It's quite common to see piggybacking effects when you have worms that have been propogating for a long time in the world."[3]

Targeted attacks are capable of inflicting much damage. In August 2000, a former employee of Internet Wire issued a false press release stating that Emulex Corporation's CEO had resigned. The company's stock plummeted. In another incident in September, the SEC brought charges against a Cedar Grove, New Jersey, teenager for orchestrating a more distributed attack, a "pump-and-dump" scheme, in which he sent hundreds of anonymous email to message boards touting selected stocks. These incidents may be the harbingers of more destructive automated attacks. Attackers have found ways to insinuate themselves into the digital financial domain and to identify subsets of victims such as careless investors. Viruses and worms piggy-backed onto these schemes may inflict more finely tuned damage.

David J. Farber, an Internet pioneer who serves on the board of the Electronic Frontier Foundation, an online civil liberties group, says that damage will increase along with dependency on the Internet. He anticipates "disinformation experts" who will plant false rumors online, wartime equivalents of the PairGain scam in which a stock speculator created a sham financial news item on a web page that looked like part of a news service's site. "There are going to be a lot of interesting experiments done in this kind of psychological warfare," he said (in Schwartz 2000).

Vigilance is a key to protection, but we must learn where to look. Public health programs for the control of parasitic infections find the weakest link in the transmission chain and break it. In complex transmission chains, parasitic diseases and viruses may be transmitted by vectors that serve as intermediate hosts of disease. In computer networks, email often performs a parallel function in the transmission of infection, serving as an intermediate host for transmission of malicious code from one computer to another. In the case of vector-borne diseases such as malaria, the most effective control programs involve control of the disease vector, mosquitoes. The disease organisms themselves are often too numerous to be controlled economically. In Table 7, we compare the methods used to control a biological vector (mosquitoes) with existing methods of control for a cybervector (email).

Control mechanisms come with associated costs, many of which are high. In the biological domain, environmental modification requires a strong vertical structure and political will. Larval source reduction programs, which were effective in eradicating *Aedes aegypti* from much of Central and South America, involved large, paramilitary organizations that enforced legislated controls prohibiting property owners from allowing mosquito production on their land. The disease burden in afflicted areas was so intense that people accepted and continue to demand these controls. Cyber control mechanisms also come with associated costs, some of which are excessive and difficult to maintain

[3] Chernobyl virus rides Klez's coattails. http://news.com.com/2102-1001-900050.html

(such as the prohibition of email). It is clear that the vertical structure necessary to implement to eradicate cyber parasites in all but severely restricted domains is impossible, given the current perceived disease burden.

Table 7. Equivalent control methods for biological vectors (mosquitoes) and cyber vectors (email)

Region of control	General method	Biological	Cyber
Vector (Induce changes in the vector population)	Genetic	Breed resistant mosquitoes	Do not use email programs that use attachments
	Mechanical	Zooprophylaxis (filter out diseased vectors)	Screen out infected emails
Host (Evade, deter, or destroy vectors coming into contact with host)	Behavioral	Stay indoors during mosquitoes' active periods	Do not read email
	Barrier	Use mosquito netting	Use filters and firewalls as a barrier to entry
	Repellent	Use insect repellent (e.g., DEET)	Employ strong security measures as a deterrent
	Mechanical	Swat biting mosquitoes	Run virus scan program to delete malicious code
Environment (Modify network structure)	Introduce biocontrol	• add a parasite of the vector (e.g., *Microsporidia*) • add a predator (e.g., *Gambusia*)	No equivalent
	Mechanical removal of vectors from environment	Spray with pesticide	Prohibit email
	Modify environment	Drain standing water where mosquitoes breed	Disable email servers during epidemics

Biocontrols have been introduced in many areas to control mosquito populations by altering community structure. *Microsporidia*, a protozoan parasite of mosquitoes, infects the eggs of infected mosquitoes. This concept of a parasite of a vector can be extended to include hyperparasites (parasites of parasites, i.e., a computer virus that infects another virus), a strategy that has not yet been implemented to our knowledge, although it is a common occurrence in nature.

For example, in July 1998, *Paracoccus marginatus*, the papaya mealybug, a parasite of papaya and cassava, was discovered in Bradenton, Florida. A subsequent

search found three parasites of the mealybug, small wasps of the same family that cause the mealybug to mummify and "blow up like a cigar." However, the researchers also found six hyperparasites infecting these beneficial wasps, which had to be eliminated from the samples prior to transport to the U.S.

3.5.2 Random mutations and evolution

Life is a paradox. Although extraordinary checks on the correctness of DNA exist, the evolution of life depends on random mutations. Cyber parasites currently incorporate extensive error-checking code. It may be possible, however, to construct malicious code that is designed to produce mutations. Survival and propagation of these mutated cyber parasites would be closely analogous to natural selection.

3.5.3 Monitoring system health with parasites

It has been suggested that parasites can serve as sentinel species, indicators of the health of an ecosystem. As we have noted earlier, the coevolution of parasites and hosts has resulted in finely balanced ecosystems. The survival of parasites relies on the persistence of complex patterns of relationships within communities. When these pathways are disrupted, populations of parasites will decline. Parasites in distress may be indicative of a troubled ecosystem.

The concept of sentinel species as indicators of ecosystem health is widely accepted. In cyber communities, it may be possible to use cyber parasites as sentinel species. The problem for computer security is how to distinguish a good virus from a bad virus. This will only become harder as the number of viruses and their level of sophistication increase.

3.5.4 Beneficial parasites

Biologists avoid classifying organisms as "good" or "bad." From an anthropocentric point of view, organisms can be damaging or helpful to man, but this perspective does not incorporate an appreciation of the structure of the community and the myriad of indirect effects that percolate through it.

Sometimes organisms must be controlled to mitigate economic damage and sometimes the cure is worse than the infestation. There are many examples of biocontrol efforts that have gone awry as introduced species choose to prey upon species other than their targets, or decimate the population of species whose importance was not understood.

The care with which the release of biocontrol agents must be planned and the frequency of unexpected consequences instills a sense of humility among biologists. In one sense, "biocontrol" is a misnomer, because control is one thing that is lacking in the management of ecosystems.

Beneficial viruses were discussed by Bontchev (1996). He listed the following arguments against putatively beneficial computer viruses:

(1) Spread cannot be controlled by the author.
(2) It may be difficult to distinguish among "good" and "bad viruses.
(3) They waste resources (disk space, CPU time, memory).
(4) They may contain bugs.
(5) The program may modify files and make them incompatible with a user's programs.

(6) Non-replicating programs may be just as effective.
(7) It is unethical to modify other people's data without their authorization.
(8) Modifying a program may violate copyright and void technical support agreements.
(9) A "good" virus may be modified and misused.
(10) Beneficial viruses condone the efforts of irresponsible virus authors.
(11) Viruses compromise the users trust.
(12) Viruses have a negative connotation.

Bontchev described some "bad" examples of beneficial viruses that demonstrate one or more of the above arguments. However, he concluded by explaining the attributes of a non-malicious, self replicating program. The presence of such beneficial viruses in the cyber landscape may be inevitable, since we may be no more successful in controlling them as we are in controlling malicious viruses. "Good" and "bad" programs are merely those that find niches to occupy among the many types of code resident in a computer network.

3.6 Summary

Ecology gives us a glimpse of the complexity of functioning, large-scale, dynamic systems. Under favorable environmental conditions, the interrelationships among the system's constituents form a framework that determines the stability or instability of the community. The survival of vast numbers of individuals of many species on Earth is a testament to the strength of natural communities. However, if communities are subjected to extreme stresses that cause them to disintegrate, some species that depend upon the community for survival are likely to become extinct.

Achieving balance in large-scale computer networks will involve management of many types of actors – users, administrators, operating systems, ISPs, programs (both legitimate and malicious) – in many, many configurations. Understanding the relationships among these actors and their cumulative effects on the community as a whole will be of paramount importance. While the survival of any one individual may be impossible to predict, the survival of the network may be assured with high probability.

Insights into biology provide an intuitive, easily understandable structure for describing malicious code. As cyber attacks involving these agents become more complex and sophisticated, an accessible classification method will be necessary to compactly communicate information about the breadth, severity, and mechanisms of attack to nonexperts. The taxonomy developed in this project provides one model from which such a classification might be constructed. Further research will be necessary to evaluate the effectiveness and practicality of the suggested classification.

3.7 References

Alberts D, Garstka JJ, Stein FP. 1999. *Network Centric Warfare: Developing and Leveraging Information Superiority.* CCRP.

Bontchev V. Are "good"computer viruses still a bad idea? http://www.virusbtn.com/OtherPapers/GoodVir/goodvir.txt (accessed 12/4/00).

Calow P. 1983. Pattern and paradox in parasite reproduction. *Parasitology* 86:197-207.

Chernoff B. 1986. Systematics and long-range ecologic research. In: Kim, K.C. and Knutson, L. (eds.), *Foundations for a National Biological Survey*. Lawrence, KS: Assoc. Syst. Coll.

Cramer ML, Pratt SR. Computer viruses in electronic warfare. http://www.infowar.com/survey/virus_ew.html (accessed 11/16/00).

Denning DE. 1999. *Information Warfare and Security*. Addison Wesley, Reading, MA.

Kohonen T. 1993. Physiological interpretation of the self-organization map algorithm. *Neural Networks* 6: 895-905.

Marcus LF. 1993. The goals and methods of systematic biology. In: Fortuner, R. (ed.), *Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, Computer Vision*. Johns Hopkins University Press, Baltimore, MD.

Morris AK, Rossignol PA. Trophic evolutionary pathways: a model based on life history parameters. Submitted.

Oka ST, Ogawa T, Oda T, Takefuji Y. 1996. A new self-organization classification algorithm for remote-sensing images. In: *Proceedings of the Adaptive Distributed Parallel Computing Symposium.* August, 1996.

Shreve J. 2000 (December). Insecurities Exchange. *Wired*.

Schwartz J. 2000. When point and shoot becomes point and click. New York Times November 12 2000: 16.

Takefuji Y, Lee KC, Aiso H. 1992. An artificial maximum neural network: a winner-take-all neuron model forcing the state of the system in a solution domain. *Biological Cybernetics* 67: 243-251.

Wilson EO, Bossert WH. 1971. *A Primer of Population Biology*. Sinauer Associates, Inc. Sunderland.

4 Epidemiology

We begin this chapter with a discussion of basic concepts in infectious disease epidemiology. We discuss previous work by others applying the concepts to computer viruses. We develop a mapping of one metric, the basic reproduction rate, from human disease to computer viruses and worms. We also develop the metrics, generation time and doubling time. We then present our results. We describe a simulation model we have developed for internal validation of the analytical model. We close with a discussion of a notional concept of operations, a description of future work and conclusions.

The work in this chapter summarizes the efforts of the Information Assurance Cyber Ecology Project in mapping epidemiological parameters to cyber threats, where these threats are modeled as diseases. The approach is ecological because the transmission depends on many 'species' within the cyber community, such as software, malware, and humans. Detailed explanations of the procedures developed and calculations performed are contained in Appendix C.

4.1 Epidemiological models of disease transmission

Disease transmission models describe the manner in which disease spreads within human populations. Epidemiologists use compartmental models that capture transition between states. Humans may pass through several disease states during the course of a disease. These disease states are often represented as acronyms, such as MSEIR, with each letter representing a class or compartment. The compartment *M* contains infants with passive immunity to infection. The compartment *S* is the class of susceptible individuals, those who can be infected by a disease organism. Infants in class *M* progress to this class when their maternal antibodies disappear. Upon adequate contact of a susceptible individual with an infective individual, the susceptible enters into the compartment *E* the class of exposed individuals. When this individual is infectious, that is, capable of transmitting the infection, he or she enters into compartment *I* the class of infectives. When the infectious period ends, the individual enters into compartment *R* the class of recovered individuals. Many possible permutations of these compartments are possible, depending upon the disease modeled and the flow patterns between compartments. Some of these models are MSEIRS, SEIR, SEIRS, SIR, SIRS, SEI, SEIS, and SI. (Hethcote 2000).

In the context of the broader environment, we can expand our focus to include disease organisms and vectors as well as human hosts. The simplest method of transmission, direct or host-to-host transmission, occurs when a disease is transmitted from one host to another without passing through an intermediate species. Only one species, the host species, is required for transmission. Time delays occur only when one host infects another. The equivalent of direct transmission is transmission of a worm directly from one computer to another, for example as in Code Red.

Indirect transmission, requiring or involving more than one species, can be mechanical or biological. Mechanical transmission occurs when the etiological agent of disease passes through one or more entities, which may be living or non-living, and are not required for transmission. Any delays that occur are incidental and not necessary for incubation. For example, cholera may be introduced into a river and it will take a certain

length of time for it to move downstream. Cholera is transmissible at the onset, and the delay is not required for transmission. Mechanical transmission occurs when the disease agent is transported through mechanical vectors, such as dirty syringes, soil, water, or even currency (such as dollar bills) and possibly some insects. The cyber equivalent of mechanical transmission is a virus transmitted on a floppy diskette.

Biological transmission can occur through one or more intermediate hosts or vectors. Disease is transmitted from intermediate hosts to subsequent hosts when the subsequent hosts incidentally consume infected intermediates or their infected by-products. Vectors transmit the disease to subsequent hosts by biting them. There is a time delay, or incubation period, between the time of infection of a host or vector and that of the subsequent host. For example, intermediate hosts play a role in the life cycle of Dicrocoelium dendriticum, the lancet worm, shown in Figure 13[4]. Snails are an obligate intermediate host and ants are infected when they ingest the slime balls produced by the snails. For viruses that spread by sending email, the emails can be thought of as intermediate hosts.

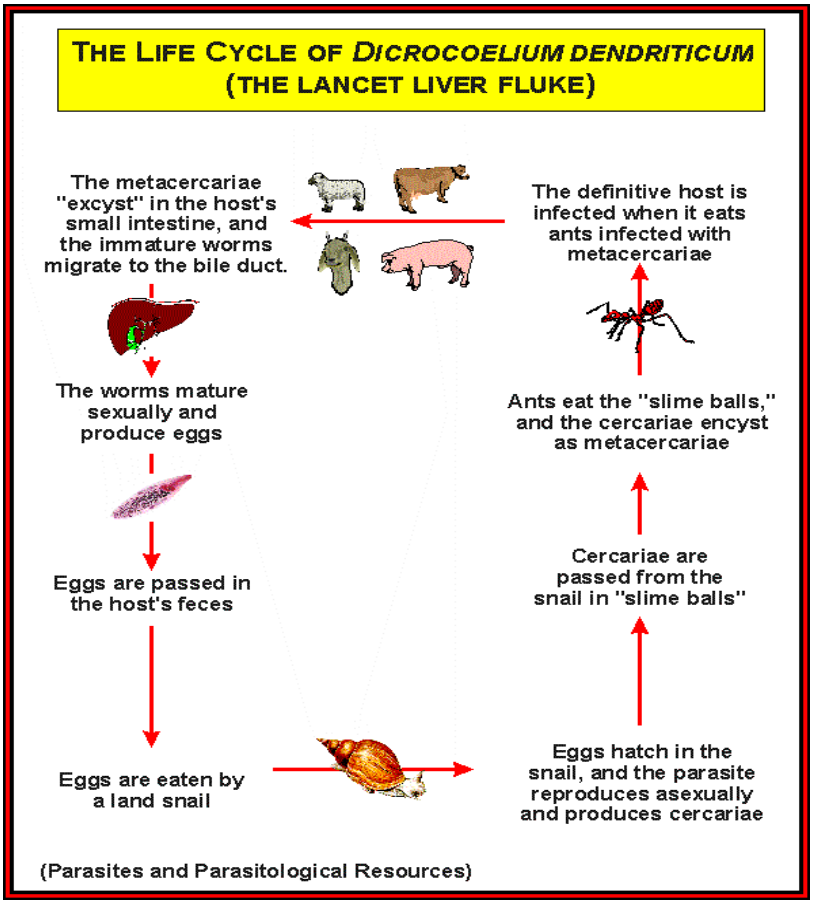


Figure 13 Life cycle of Dicrocoelium dendriticum

[4] http://www.biosci.ohio-state.edu/~parasite/lifecycles/dicrocoelium_lifecycle.html

In vector transmission, a vector is typically capable of infecting multiple hosts (Figure 14). There are at least two incubation periods, in the vector, in the host, and in any other intermediate species. A pest acquires the etiological agent, and then transmits it to the host organism. Vector transmission occurs in some Microsoft Word macro-viruses, where an infected document infects a template, which in turn is capable of infecting many more documents. Vectors are an important consideration in disease transmission because they dramatically increase the potential for spread. A vector facilitates transmission to many hosts much more suddenly than is possible with direct transmission. If we consider emails to be vectors then an email to a large mailing list can spread a virus to many hosts simultaneously.

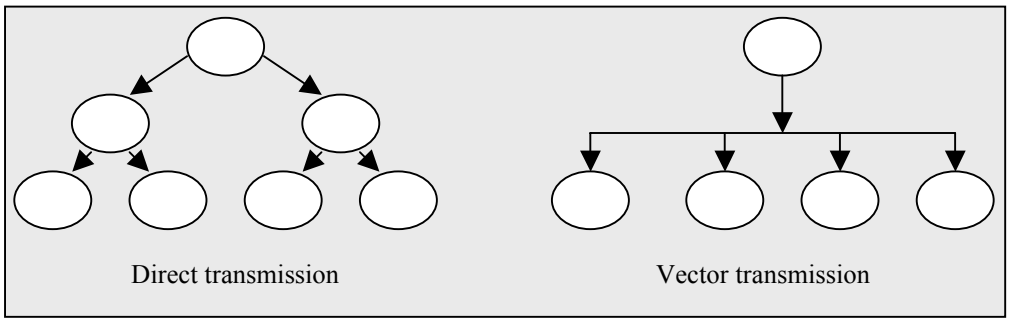


Figure 14. Patterns of direct and indirect transmission

Transmission through vector and intermediate hosts implies the existence of a community. For example, if sheep, snails and ants were not frequently found in the same meadows *dicrocoelium dentriticum* could not survive. Similarly, Microsoft Word macroviruses depend on the association of Microsoft Word documents and templates, and email viruses depend on the association of computers and emails. We can reasonably consider these associations to be communities.

These theoretical concerns have minimal impact upon the practical monitoring of the spread of malicious code in human time. The viruses are so efficient, and transmission, whether analogous to biological, mechanical, or direct transmission, occurs so quickly, that complex disease transmission may be modeled effectively using simple models. Time delays become important when they represent human contributions to the spread of cyber disease, such as the time between receipt of an email by a host computer and execution of an infected attachment by the human recipient of the email.

4.2 Previous work

Several individuals have modeled particular viruses or worms on an *ad hoc* basis and have produced estimates similar to the results presented in this report[5]. A goal of the Cyber Ecology project is the systematic development of broadly applicable procedures for the rapid generation of useful models.

[5] An excellent example of this is Stuart Staniford's analysis of the Code Red worm published on the CyberPanel community mailing list (cpc@schafer-ballston.com) on 20 July 2001.

Previous published work in cyber epidemiology consists primarily of a series of technical reports by Jeffrey O. Kephart, Steve R. White and David M. Chess at the IBM Thomas J. Watson Research Center that examine the problem of computer viruses from an epidemiological perspective. Overall, the authors (mostly Kephart and White) make the point that computer viruses have a pattern of spread that is 'biological', taking into account some topological considerations. They argue that the focus so far has been at a micro-level, typically code. A macro-outlook is lagging but necessary with the continuing spread of computer viruses. They attempt to lay the theoretical foundation of this new emerging science.

Their first paper, *Computers and Epidemiology* (Kephart, et al. 1993) simply lays out the premise and defines terms from epidemiology and population dynamics, explaining a basic model of disease spread. They emphasize the concept of epidemic threshold as a target for control, maintaining that one need not detect every virus to stop its spread. Topology is then addressed in detail, analogous to dispersion patterns in epidemiology.

Computer Viruses: A Global Perspective (White, et al. 1995) examines the spread of viruses worldwide. They make some important practical points on data collection, notably the concept of virus incident, which is the beginning of an infection within a particular topological unit. They also make a novel point, which is that the number of new virus species is *not* growing exponentially, but roughly linearly. The parameters explaining the comparative prevalence, present or future, of different viruses are unknown. They compare various viruses, both file and boot viruses, the last being expected to increase with networking. The misconceptions of the media are examined with the Michelangelo virus.

Directed-Graph Epidemiological Models of Computer Viruses (Kephart and White 1991) is a modeling effort to support the concept and potential use of epidemic threshold.

How prevalent are Computer Viruses? (Kephart and White 1992) analyzes Dataquest and CertUS data. They further extend the biological/ecological analogy, and introduce immunology and preventive health. They underline some major deficiencies in our birth-death rate models of spread and equilibrium, in that none satisfactorily explain (and therefore do not predict) the discrepancy in prevalence levels of surviving viruses. Again, they imply that the answer lies in topology. They state that *worms* are probably the greatest threats in the future, because of their ability to spread around the world in hours. They are eventually eradicated, but every worm is a new worm. Detection of worms is fraught with problems because they spawn through active processes, a legitimate activity. They suggest that examining collective behavior of many systems may help but, overall, they are skeptical and express a tremendous concern over the future threat from worms.

Measuring and Modeling Computer Virus Prevalence (Kephart and White 1993) follows up on the hierarchical concept of topology. A model suggests that a 'kill signal', hunting for viruses in neighboring machines, would be very effective in stopping spread. They advocate a proactive rather than reactive anti-virus programs.

Overall, this prior literature indicates that the epidemiological analogy is valid and explains the general behavior of viruses. However, the authors indicate that differences between the viruses, such as among equilibrium levels, are not explained by this approach. We propose that a model of higher organization, of community rather than

population, may provide insight into these differences. Community models use population level parameters, but integrate them with community-level effects. Complex relationships will result in counterintuitive changes in levels and turnover rates within a community that are not necessarily predictable from a life table approach.

More recently, Pastor-Satorras and Vespignani (2001) have simulated the epidemic spread of computer viruses in scale-free networks. Using prevalence data, they show that many computer viruses persist at endemic levels over long periods of time, without exhibiting the property of an epidemic threshold. This finding that sufficiently large, scale free networks support endemic levels of disease highlights the importance of monitoring rates of spread. We develop metrics based on those used in public health monitoring to assess the rate of spread (basic reproduction rate, generation time and doubling time) in this report.

4.3 The basis of infectious disease epidemiology: the Ross model

Infectious disease epidemiology is rooted in the work of Ronald Ross. At the turn of the twentieth century, Ross elucidated the life cycle of malaria and showed that mosquitoes were an obligatory vector for the parasite. His subsequent "theory of happenings" revolutionized infectious disease epidemiology and malariology. His theory described the importance of "dependent happenings" in the transmission of infectious disease, the occurrence dependence of infectious disease in individuals upon the occurrence of disease in other members of the population. Ross's model incorporated both disease organism and host. Since infectious disease epidemiology concerns the relationships among disease organisms, host organisms, and the environment, many epidemiologists consider it to be an extension of the science of ecology (Halloran 1998).

Ross devised the model to answer a specific question. Having gained the knowledge that mosquitoes are vectors of the disease, Ross wished to know whether mosquito eradication was a prerequisite to malaria eradication. This question was most troubling because it was known even at that time that mosquito eradication was unrealistic on a broad scale.

The model calculates the basic reproduction rate (BRR), R_0, of malaria. By definition, R_0 is the mean number of infective secondary cases generated by a primary case over its average duration. The derivation can be achieved a number of ways, the first here being an algebraic one devised by Macdonald.

Assuming that hosts and vectors are uniformly distributed and that no cases currently exist, one can calculate the number of cases that one primary case would generate. First, there are a certain relative number of vectors, *m*, which is the number of vectors divided by the number of hosts. Given that female mosquitoes (males do not blood feed) only bite once per period of egg development and that only a certain proportion will bite a human host when doing so, only a proportion, *a*, the biting habit, will actually bite on a given day. It consists of the proportion biting human beings divided by the length of the oogonic cycle. The product, *ma*, is then simply the number of mosquitoes biting one host per day, and is called the man-biting rate. Once the parasite is acquired, it must undergo a long extrinsic period of incubation, of duration *n* days, before the vector can infect a host. The duration of incubation is substantial, and often longer than the average life expectancy of the vector. On any day therefore, only a proportion, *p*, of vectors survive, and thus only a proportion, p^n, survive the incubation

period. These infective vectors still bite at a daily rate, a, for $-ln(p)^{-1}$ days. Their efficiency of transmission is labeled b. This represents then the proportion of human cases generated per day on host infection, which must then be multiplied by the duration of infection, r^{-1}, where r is the recovery rate.

The product of these parameters represents the basic reproduction rate of malaria, namely,

$$R_0 = \frac{ma^2p^nb}{r(-\ln(p))}$$

The basic reproduction rate must be equal to unity for malaria for maintain itself, and therefore must be less than one for extinction to occur. Given that none of the parameters are 0, it can be seen that the number of vectors, m, need not be reduced to 0 for eradication to be successful.

Three unexpected properties emerged from the model. First, there is a non-zero threshold for all parameters. Second, there is a hierarchy of parameters, with m being linear while a is square, and p is exponential. Third, it is possible to obtain a linearly proportional estimate of the basic reproduction rate in the absence of the parasite and from entomological parameters only. This practical tool was used extensively during eradication campaigns and is known as vectorial capacity (*C)* (Bailey 1982), namely,

$$C = \frac{ma^2p^n}{-\ln(p)}$$

Furthermore, the model is notable in that it incorporated a time-delay (the extrinsic incubation period), still a difficult aspect of modeling, in an elegant and ingenious fashion. The model was nothing short of extraordinary. It provided a theoretical basis for understanding transmission, pointed to specific methods of measurements and estimation, and managed to incorporate difficult modeling problems. For these reasons, it has remained a source of study to this day.

4.4 Modifying the Ross model for email viruses

In the cyberworld, the attribution of parameters to host and vector differs from that in the description of biological disease. In Ross's model, r and b were contributed by the host and m, a, p, and $-1/ln(p)$ were contributed by the vector. In the biological model, m represents the ratio of vectors to hosts; it is the total number of potential contacts available to one vector. In the transmission of email viruses, m is functionally equivalent to the average size of Outlook address book. The man-biting rate appears twice in the Ross model, as the probability that a vector will bite prior to and following infection. In the cyber model of email transmission, two probabilities are also involved in transmission, but they are distinct. The probability that the virus will bite can be thought of as the probability that the virus will infect emails sent from the infected computer. This process is very efficient and near unity. The second transmission probability is anthropogenic, the probability that the human recipient will execute an infected attachment. This parameter contains significant variability and is affected by the user's interest and awareness of potential infection. Since the transmission probabilities are not

completely determined by the behavior of the virus, the metric analogous to vectorial capacity cannot be derived for cyber email viruses that require execution of an attachment. The resulting quantity,

$$R_0 = ma$$

describes the short-term reproductive potential of a perfectly efficient disease agent that is transmitted to all contacts in a completely susceptible population without recovery. While this model is unrealistic biologically, it is very descriptive of the spread of email viruses through executable attachments. Email viruses, such as Kak, that do not spread by executable attachments, evade detection and are able to persist on the host. We add the life span parameter to the simple email model to capture this behavior.

The transmission parameters used to calculate BRR can be summarized in a life cycle diagram. The life cycle diagram is a graphical representation of the transitional states of a particular virus or worm. These states are abstracted from technical descriptions of the virus or worm, for example, the summaries published on the World Wide Web by anti-virus companies. Variables describing attributes such as the size of the address book, are multiplied by probabilities, such as the probability of executing an infected email attachment, to derive BRR. The life cycle diagram for the email virus, Anna, is shown in Figure 15. We estimate Anna's BRR to be 0.7 * 70 * 0.1 = 4.9.

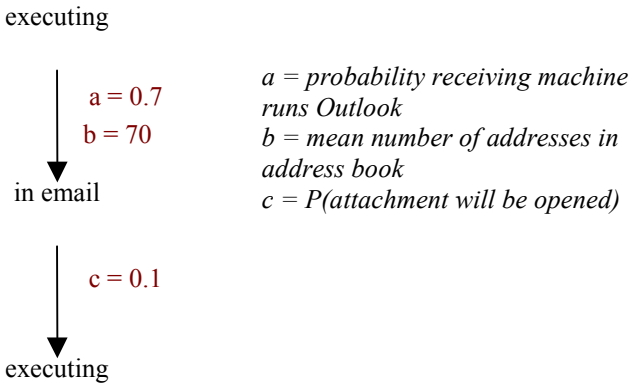


Figure 15. Life cycle diagram and transmission parameters for Anna

4.5 Practical application of BRR

BRRs are used in public health to assess the epidemicity or endemicity of a contagious disease. It is often considered to be a threshold quantity that determines when an infection can invade and persist in a new host population. When the BRR is unity, a stable state exists where each infection produces on average one additional infection. The disease is endemic, that is that it persists at a constant, often low level in the host population. BRRs less than unity indicate that the disease is in decline. BRRs greater than one indicate that the infection is growing.

BRR provides a method for comparing different diseases. It is used in public health as a practical tool for evaluating the effects of intervention measures, such as the application of insecticides, to control disease. Field models are used by public health workers on the front lines of the battle against infection, to monitor disease state during a

control campaign. BRR may be calculated on a village-by-village basis and used to tailor control strategies in a localized manner. Careful, small-scale experiments were and continue to be conducted to assess incremental changes in mosquito density, life expectancy, and biting rate to determine the effects of control regimens on model parameters. Changes in BRR provide a rapid index that enable experienced workers to quickly assess a situation and determine the severity of a disease outbreak and the effects of control measures.

In order to produce comparable BRRs for infectious malicious code we use default parameters. We feel these parameters are reasonable first estimates. One way to make the BRRs more precise would be to collect information about these parameters in the real world. We suggest that this is one way for a network administrator to assess the vulnerability of his network. It is also a way to measure incremental improvements in defensive measures.

4.6 Construction of BRR for malicious code

The pivotal contribution of the Ross model is the marrying of patterns of contact with transmission probabilities. BRR is built on state transitions. Analogously to the assessment of BRRs by public health professionals in human populations, we assess the BRRs of viruses and worms in populations of computers by identifying the transition states of the infection and associating quantities and probabilities with these transitions. We have found that life-cycle graphs provide an intuitive, visual method for describing state transitions and time delays. Life-cycle graphs for the malicious programs analyzed in this report are shown in the Appendix C.

Although we calculate the value of BRR deterministically, the actual occurrence of many parameters is most certainly stochastic. For example, the number of recipients per email is stochastic with an unknown, but highly skewed distribution - many emails are intended for only one recipient, but others can have numerous recipients. The parameter value we use is our estimate of the geometric mean of this distribution, which is five.

Some state transitions that occur with probability equal to one are not called out separately in the model. For example, the probability that a computer running Windows will require a reboot is not given since rebooting is a fact of life for Windows users.

For some of the parameters, default parameters are difficult to assign because the probabilities vary from case to case. For example, the probability that a recipient will open an email attachment depends upon:

- the number of identical messages received;
- whether or not the sender is a trusted source;
- the message line and whether the infection can be recognized from this information;
- increased awareness of viruses and worms because of media reports.

All of the examples of email viruses we have studied require Windows and Microsoft Outlook. We have built our estimates under the assumption that 70 per cent of senders and recipients of email use this software combination.

Infectious spread for many email viruses depends on the recipient not deleting the email on arrival. We place this probability at 0.6. Then, the recipient must open the attachment and the virus program must execute. We place the probability of this event at 0.1. This probability may be significantly reduced by the presence of anti-virus

definitions on the recipient's computer. This may result in significant variability in BRRs. This is a parameter that a system or network administrator may be able to estimate for his or her network to obtain a more precise estimate.

Default parameters for calculating BRR of an email virus are:

- proportion of computers using Outlook and Windows .7
- mean number of addresses in an Outlook address book: 70[6]
- probability that email attachment is executed: 0.1

We analyzed a worm, Kak, and found that two additional parameters were necessary to describe its transmission cycle. Kak spread relatively slowly relative to the other infections we analyzed and was able to contribute multiple infections over time, prior to detection. In addition, it piggybacks on legitimate email messages sent by users rather than sending its own. The parameters added to describe these behaviors were:

- mean number of recipients per email: 5
- mean number of legitimate email messages: 21
- probability that recipient does not delete file on arrival: 0.6

Many infectious cyber agents spread through multiple modalities. LoveLetter, for example spread not only through email, but also through mIRC and by replacing many types of files such as .jpgs and .mp3s, with infected copies. Additional default parameters are required to calculate the BRR for these agents. Each transmission modality is treated as a separate branch of the life cycle. One branch, such as email transmission, may account for the initial epidemic rise of the infection, while another branch, such as mIRC or in file transmission may contribute at a lower level over time. The effect of this continued, low level of transmission is to maintain an endemic level of infection in the susceptible population for a period of time exceeding the duration of the initial infection in length. In the BRR calculations, we call out the transmission specific and total estimates for BRR.

Default parameters contributing to the BRR of mIRC transmission are:

- probability that mIRC is used on a computer. We set this probability to 0.1 in our example calculations.
- mean number of IRC users on a channel at a given time. We set this to 20.
- probability that a recipient does not delete the infecting file on arrival. We set this at 0.6.
- probability that a recipient runs this file given that it was not deleted. We set this at 0.4.

Additional default parameters are also needed to describe in file transmission:

[6] Our confidence in this estimate has been reinforced by a recent small survey conducted by Scott Musman at IMSI. Discarding outliers, the mean size of address books he observed was within 20% of this estimate. (Personal communication 01 Aug 2001).

- mean number of files overwritten. For LoveLetter, we set this to 300.
- probability that any given file will be transferred to another machine. We set this to 0.1.
- probability that a recipient does not delete the infecting file on arrival. We set this at 0.6.
- probability that a recipient runs this file given that it was not deleted. We set this at 0.4.

BRR represents one aspect of the spread of disease that describes infectious spread. It incorporates parameters such as the average number of contacts, probability of transmission, and duration of contact. Only those parameters that pertain to replication are included in the calculation of the BRR. Confidentiality attacks, resulting for example from transmission of a Trojan horse, or denial of service attacks resulting from mass mailings, are not addressed by the BRR.

Cyber infections spread by execution of email attachments require intervention by the user, who must click on the attachment to execute the virus program. These viruses are relatively obvious and control is relatively straightforward. Media coverage often alerts large numbers of users to the presence of the virus and/or worm and control measures are publicized. The simplest control measure is to not click on these attachments.

Worms that spread automatically, such as Kak, are another matter. They may elude detection for long periods of time and continue to infect over an extended period of time. For these worms, we do account for additional cases infected over time in the BRR calculation. That is, BRRs will generally tend to be high because of an extended period of infectiousness.

We analyzed one macrovirus that spread through infected Microsoft Word documents, Ethan. Ethan then infects the Microsoft word template, so that all documents processed in Word are subsequently infected. The virus spreads to other documents on the same computer, and to documents on other computers when these documents are transferred between computers. We examine the case where infected documents are attached to email and opened on the recipient's machine. The parameters that capture this behavior are:

- mean number of uninfected documents on a host: 25
- probability that recipient does not delete email on arrival: 0.5
- mean number of legitimate emails sent: 300
- probability that the email contains a Microsoft Word attachment: .05
- probability that the file sent is infected: 0.9

Life cycle graphs, basic reproduction rate, generation time and doubling time calculations for all the malicious code analyzed are shown in the Appendix.

Other aspects of disease, complementary to BRR, are morbidity and mortality. Virulence is a measure of the speed with which a parasite kills an infected host. In infectious disease epidemiology, this information is expressed as the case-fatality ratio, the probability of dying from a disease before recovering or dying of another cause. Among cyber infections, the virulence of an infection can be inferred from the level of resulting damage.

BRR is generation based, since it represents the mean number of infected individuals that result from one infected individual. To explore the temporal aspects of spread, such as the number of cases expected within a given time interval, it is necessary to quantify

generation time. Epidemics are characterized by an initial sharp rise in the number of cases. The ideal shape of this curve is shown in Figure 16. The sharpness of the epidemic curve indicates the speed of attack. One metric that captures this information is doubling time. The total number of cases may be reduced by countervailing forces, such as anti-virus definitions and increased public awareness.

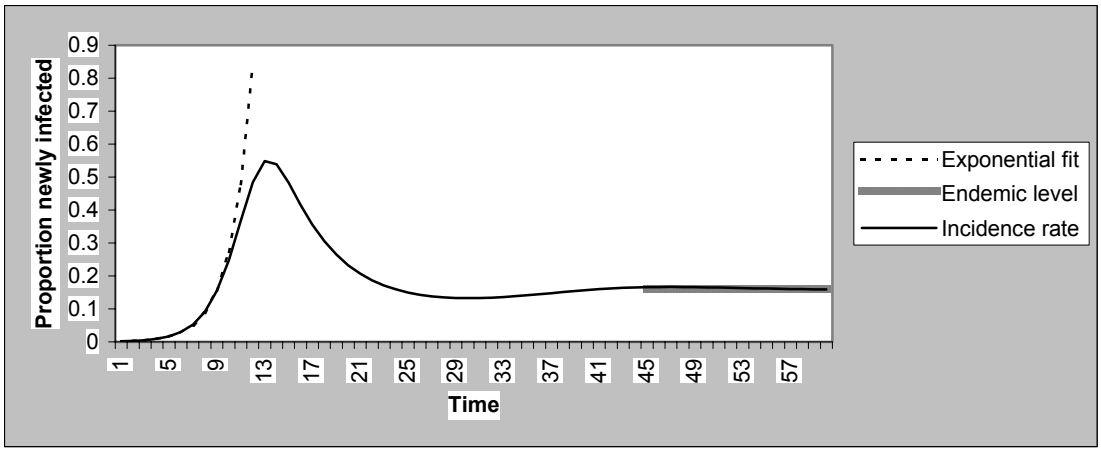


Figure 16. Epidemic incidence curve

Generation time is computed from delays between events. Biologically, generation time is the length of time between reproducing females. We define generation time for computer viruses and worms to be the mean length of time between initial execution on two hosts, one of which has infected the other. For computer viruses and worms, generation time is built from the mean lengths of delays between executions. For example, if a computer must reboot before a virus program may execute, the mean period between delays is incorporated into the generation time for that virus. When multiple transmission paths are present, we calculate the generation time for each path independently.

Some of the delays we have used to construct generation times for the examples in this report are:

- Mean time to reboot – We make the assumption that a computer using the Windows operating system will need to reboot every 12 hours. Assuming a uniform distribution over this 12 hour period, the mean time between a prerequisite event (such as infection) and the next reboot is 6 hours. This applies when the virus requires a reboot to execute.
- Mean time until sending email – We assume that email is not sent continuously, but in spurts or batches every 8 hours. Assuming a uniform distribution over this 8 hour period, the mean time between a prerequisite event and the next time email is sent is 4 hours. This delay applies when the virus attaches itself to legitimate email messages.
- Mean time between email being sent and read – Likewise, we assume that email is not read continuously, but periodically. We assign a mean period of 6 hours between when email is sent and when it is read. While many emails are read

shortly after they are sent, many others are sent during times when the user is not likely to read them promptly, such as in the middle of the night.

Generation time may be calculated by associating delays with the state transitions captured in the life cycle diagram. The generation time for Anna is six hours and is visually illustrated in Figure 17.

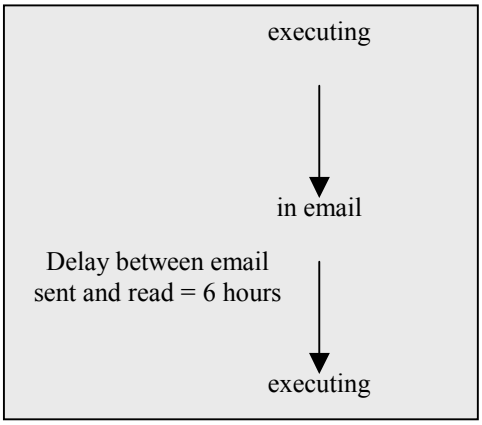


Figure 17. Generation time for Anna from the life cycle diagram

Using generation time and BRR, we calculated doubling time using the formula:

$$\text{doubling time} = \text{generation time} / \log_2 \text{BRR}$$

This formula is exact when generation time is a fixed constant. When generation time is variable, the result is only approximate. We estimate the doubling time for Anna to be 2.6 hours.

4.7 Reconciliation of data

We have calculated the BRR, generation time, and doubling time for six examples, three email viruses (Pretty Park, LoveLetter, Anna Kournikova), a email worm (Kak), a hybrid virus/worm (MTX), and a macro virus (Ethan). For each, we perform the following analysis:

- Analysis of technical description for state transitions and delays;
- Calculation of BRR. We compare these with BRRs derived in public health to describe the potential for spread of human diseases;
- Calculation of generation time;
- Calculation of doubling time;
- Estimation of time between release and peak number of infections, when published estimates were available.

We also present the graphs of actual reports received by two major anti-virus companies. The two companies use very different methods for disinfection and have geographically separate customer bases. Anti-Virus Company 1 intercepts email prior to

receipt at scanning towers. The email is scanned, disinfected if necessary, and forwarded to the intended recipients. Anti-Virus Company 2 receives reports of infection when users initiate a virus scan. These reports are user-driven. Discussions with an analyst from Anti-Virus Company 2 revealed that the numbers of reports of all viruses often spike simultaneously following media reports about any virus, because these reports lead many users to initiate virus scans. Other patterns in self-initiated virus scans include a spike after Christmas and in the autumn at the start of the school year, and a decrease in the summer. Because the data from Anti-Virus Company 2 contain fluctuations which do not reflect activity of the target virus, they are less useful for deducing incidence trends than data collected by scanning towers.

Our models behave favorably against published reports and observed behavior in the data provided, particularly considering the vague estimates assigned to the transmission parameters.

The estimated doubling time for the email form of LoveLetter is 3.6 hours. Published reports estimated the total number of opened attachments at 1.9 million is one day (Kelsey 2000). Using the calculations in our model, a total of 1.9 million infections would be achieved along the email transmission branch between 7 and 8 generations (25.2 – 28.8 hours).

The estimated doubling time for Anna is 2.6 hours. One million infections would be achieved between 8 and 9 generations (20.9 – 23.4 hours). Published reports during the outbreak indicated that the virus was spreading "twice as fast as the Love Bug" (Shipp *in* Leyden 2001). This is inconsistent with our estimates. Even though the generation time was less, the concommittant low BRR in our estimates led to one half as many total infections over a 24 hour period.

The estimated doubling time for Kak is 2.2 hours. This estimate disagrees with real world data that show very slow growth. Kak was found in the wild in late 1999, but never achieved explosive growth. Anti-virus companies did not list it as a top threat until Summer 2000, after an incident in which it was mass mailed to 50,000 email addresses (Sullivan 2000). Our estimates do not account for the effects of counter measures, such as patches and anti-virus signatures. We speculate that signatures that counter Kak were in place very soon after Kak's release. Such signatures may even have preceded Kak's release because Kak exploits a vulnerability that was previously used by another worm, Bubble Boy. Kak may be an example of effective defense against a precocious worm.

The estimated doubling time for MTX is 41 hours. This is approximately half the doubling time of seven to eight days shown in the data obtained from Anti-Virus Company 1. This doubling time is based on a BRR of 1.24, which is very close to the BRR of 1.0 indicating control.

In most of the example viruses we studied, real-world data did not capture the initial outbreak. Instead, we have observed fluctuations in the endemic phase of disease in the population of computers. Control in this phase is also important. Relaxed vigilance may result in a resurgence of disease.

4.8 Results

The parameter values used to calculate the basic reproduction rates of PrettyPark, LoveLetter, Anna, Kak, MTX and Ethan are shown in Table 8. The procedure for

calculating generation time, basic reproduction ratio and doubling time for each virus or worm is illustrated in detail in Appendix C.

Table 8. Summary of Results: Parameter values, generation times, basic reproduction ratios, and doubling times for PrettyPark, LoveLetter, Anna, Kak, MTX and Ethan

Name	Branch	a	c	d	e	f	g	h	l	j	k	l	m	n	q	r	Generation Time (Hrs)	BRR	Doubling Time (Hrs)
PrettyPark	email							70	0.05	0.7							6	2.45	4.6
LoveLetter	file	300	0.1	0.6	0.4					0.7							726	5.04	
	irc			0.6	0.4	0.1	20			0.7							198	0.336	
	email							70	0.2	0.7							12	9.8	3.6
Anna								70	0.1	0.7							6	4.9	2.6
Kak				0.6					1	0.7	5	21					12	44.1	2.2
MTX	email								0.07	0.7	5	10	0.5				12	1.225	41.0
	file		0.1							0.7						0.1	1440	0.007	
Ethan	same host									0.7				25			6	17.5	1.5
	new host			0.6						0.7		300			0.05	0.9	72	5.67	28.8
	(via email)																		

Parameter description

During one generation:

a=mean number of files overwritten	l=mean number of legitimate email messages
c=P(file being transferred to another machine)	m=P(virus does not exit due to finding anti-virus software)
d=P(recipient does not delete file on arrival)	n=number of uninfected documents on a host
e=P(recipient runs file given it was not deleted)	q=P(email contains Microsoft Word attachment)
f=P(mIRC is used on a machine)	r=P(sent file is infected)
g=mean number of IRC users on a channel at a given time	
h=mean number of addresses in address book	
i=P(attachment will be opened)	
j=P(use Outlook and Windows)	
k=mean number of recipients per legitimate email	

These threats can be ranked by the magnitude of the basic reproduction rate. This ranking shows the relative level of threat posed by each threat in terms of potential for rapid spread.

In Figure 18, we show the BRRs calculated for the examples of malicious code analyzed in this report against the BRRs for some common infectious human diseases. The BRRs for the more vector-like malicious agents, Kak and Ethan, have the highest BRRs of the viruses and worms studied. However, very high BRRs, for example on the order of the BRR of 300[7] for malaria, were not observed in this subset.

The BRR provides a proactive metric for describing the potential for spread of viruses and worms. It may provide a useful metric for planning and resource allocation. We may increase its fidelity by incorporating more accurate information about the human-driven aspects of transmission and countermeasures. These may be highly variable as evidenced by the disparate epidemic trajectories of Anna and Pretty Park, which may be due in large part to the differential allure of the purported attachment contents.

[7] Molineux et al. (1978) report values for the parameters of malaria transmission in sub-Saharan Africa that allow us to estimate basic reproduction rate (BRR). From an entomological perspective, BRR is the product of vectorial capacity (*C*), duration of infection (*1/r*) and efficiency of transmission (*b*).

$$\text{BRR} = \frac{Clb}{r}$$

From the tables in the study, the maximum BRR can be up to approximately 500, which is the number of secondary cases potentially generated from a primary case over its infective period. Similar values can be reached from inoculation rates (which are parasitological rather than entomological). While malaria has reputedly the highest BRR of any human disease, BRR is dimensionless in that the denominator of the rate is generation time, not real time. Comparisons of BRR values must be made with this fact in mind. A study by Rosenberg et al. (1990) reports parameters of BRR in Thailand that, from our calculations, reach over 100 in value.

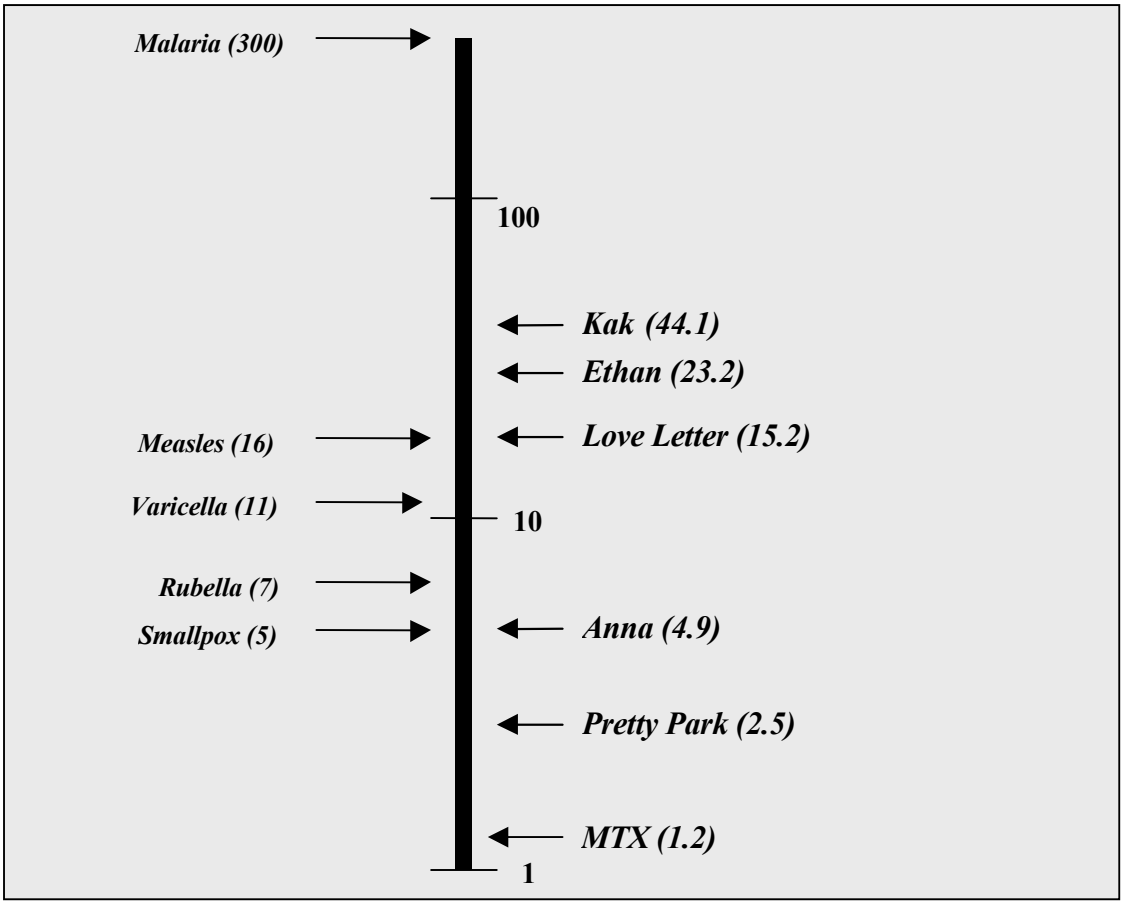


Figure 18. Comparison of BRRs for human and cyber disease

4.9 Sensitivity analysis

We performed a sensitivity analysis to assess the stability of the calculations. We varied the parameters, singly and in combination, and observed any change in the results.

In Figure 19, we observed the effect of changes in the mean number of addresses in the Outlook address book on doubling time. The resulting curve depicts the expected behavior, an inverse relationship. Doubling time is sensitive to the number of addresses when the address book contains fewer than 50 addresses. Doubling time decreases sharply to this value and then levels out.

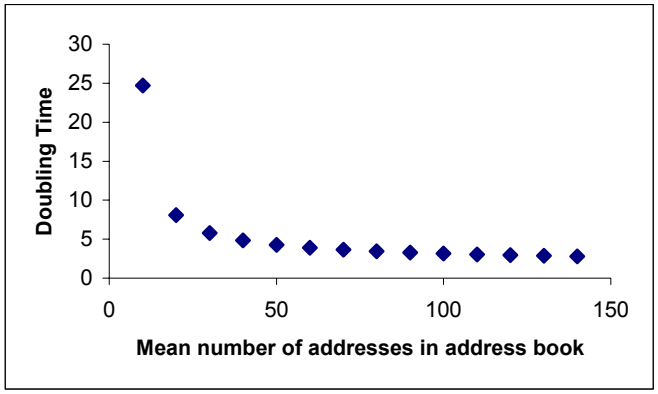


Figure 19. Mean number of addresses vs. doubling time

In Figure 20, we graph the mean number of addresses in the Outlook address book against BRR. The resulting curve is a linear, increasing function. The number of addresses is directly correlated with BRR. We observe a similar pattern in Figure 21, Generation Time vs. Doubling Time.

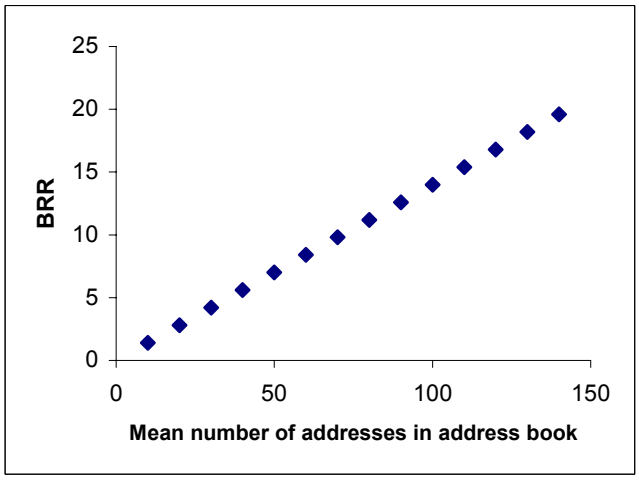


Figure 20. Mean number of addresses vs. BRR

To visualize the effect of changes in the number of addresses and generation time on doubling time, we used a three-dimensional surface plot (Figure 22). The graph displays a waterfall pattern. The graph implies that to control doubling time, it is necessary to restrict both the size of the address book and generation time.

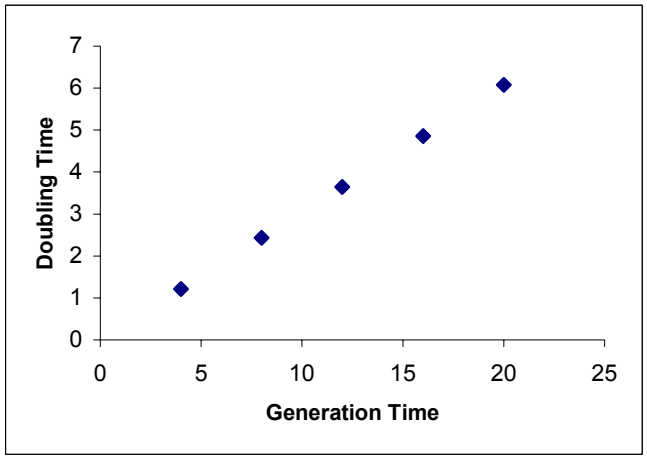


Figure 21. Generation time vs. doubling time

4.10 Simulation

We now present an individual-based ecological simulation of viral spread for an email virus. We present this simulation as a tool to be used by system and network administrators, analysts, and decision-makers to monitor the vulnerability of their networks, much as field studies are conducted by public health professionals to monitor the force of disease among human populations.

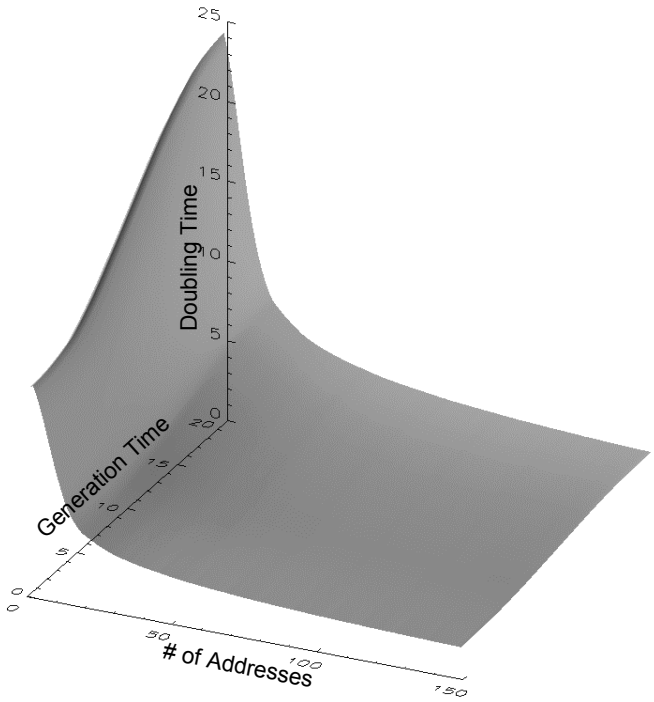


Figure 22. Number of addresses and generation time vs. doubling time

The individual-based model simulates the viral spread of an email virus that spreads via email attachments. When an infective attachment is executed, the virus first replicates itself, generating many copies of itself in the victim's computer. Then, it sends itself to everyone listed in the victim's address book (here assumed to be the Microsoft Outlook address book). In this manner, the virus can replicate itself extremely quickly. There are three modeled populations in this model: computers, email viruses, and anti-virus software. Each member of a population and its interactions with other members are simulated individually.

The purpose of the simulation program is to provide internal validation of the analytical model. It is a simulation program constructed written in PV Wave, a graphically based programming language. The simulation captures the dynamic spread of infection with simultaneous variation in multiple parameters. The complex behavior of the analytical model could not be captured in the sensitivity analysis.

The simulation program is not included in this report, but will be submitted in a future deliverable as part of the Cyber Ecology Toolkit. We submit a summary of the simulation program here to document the parameters included in the internal validation.

4.10.1 Computers

Each computer has a limited set of parameters that indicate the properties of that computer:

- ID. A unique ID number represents each computer.
- Location. (x, y) coordinates represent each computer's location in the cyber space.
- Infection status. Non-infected, infected, infective, immune.
- Number of email addresses. Number of email addresses in the address book, which is a random number following a lognormal distribution.
- Number of files over written by the virus. A random number following a lognormal distribution.

4.10.2 Virus

Each email virus possesses the following attributes:

- ID. A unique ID number.
- ParentID. ID of parent virus.
- Host computer ID. ID of computer on which the virus resides.
- Activity status. Active or inactive.
- Execution status. Yes or no
- Execution time. Discrete time steps after the virus is created following a Poisson distribution.

4.10.3 Anti-virus

Currently, anti-virus is not a simulated agent. It is a trigger that switches immune status from infective (or infected) to immune. Trigger activity is based on user-defined probability distributions.

4.10.4 Interaction parameters

Interactions within the simulation model are determined by probabilities:

- P_d. Probability that recipient does not delete file on arrival.
- P_b. Probability that the attachment will be opened before the virus alert.
- P_a. Probability that the attachment will be opened after the virus alert.
- N_a. Number of files overwritten (follows a lognormal probability density function).
- N_h. Number of addresses in address book (follows a lognormal probability density function).

4.10.5 Environment

To facilitate visualization of the cyber space, we restrict it to two dimensions in the simulation. Computers are represented as distributed uniformly across a two-dimensional array. Each computer is represented as a particular (x, y) location in the array. It is important to note that positions in this two-dimensional space are essentially random and not spatial locations of computers, and that the geographic neighbors around a computer do not imply networking connections.

4.10.6 Simulation description

The simulation is performed through a series of discrete time steps corresponding to hours. This time frame is short enough to model epidemic behaviors, specifically epidemic rise. The computer population size is 256 x 256; i.e., each computer occupies a location in a 256 x 256 grid. In the beginning of each simulation, one hundred viruses are created and each virus is randomly assigned to a host computer. During any given time step, each virus will respond individually to initialize a new viral focus of spread through the email list in the victim's computer.

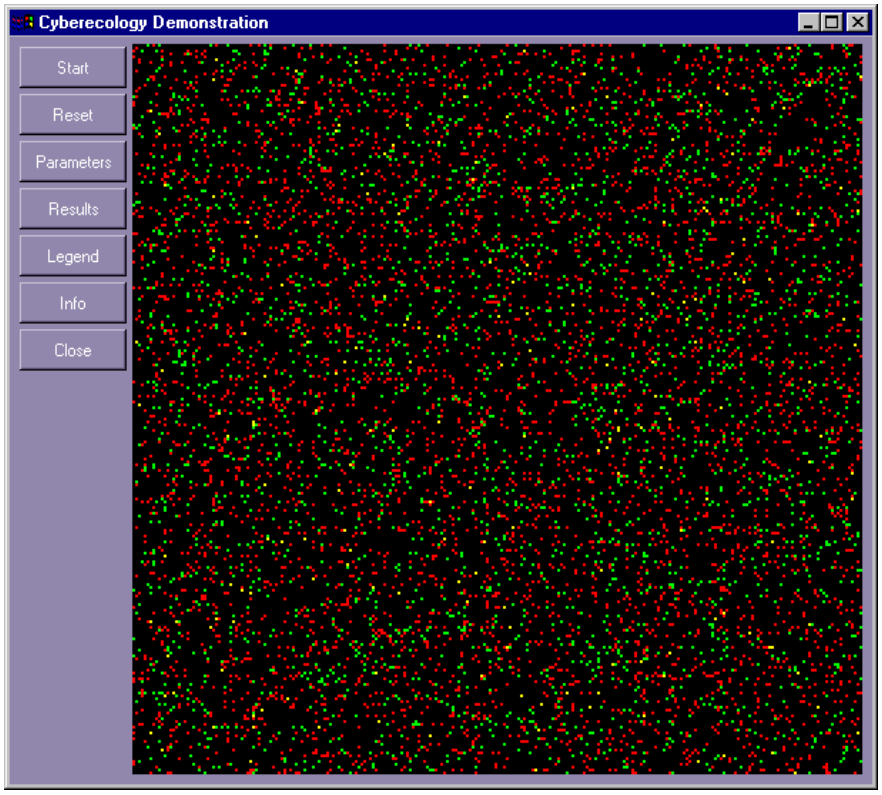


Figure 23. Screen shot of simulation interface for visualization

The simulation interface for visualization is shown in Figure 23. The main menu appears as buttons along the left side of the window. The states of the computers included in the simulation are shown in the central window as they change over time.

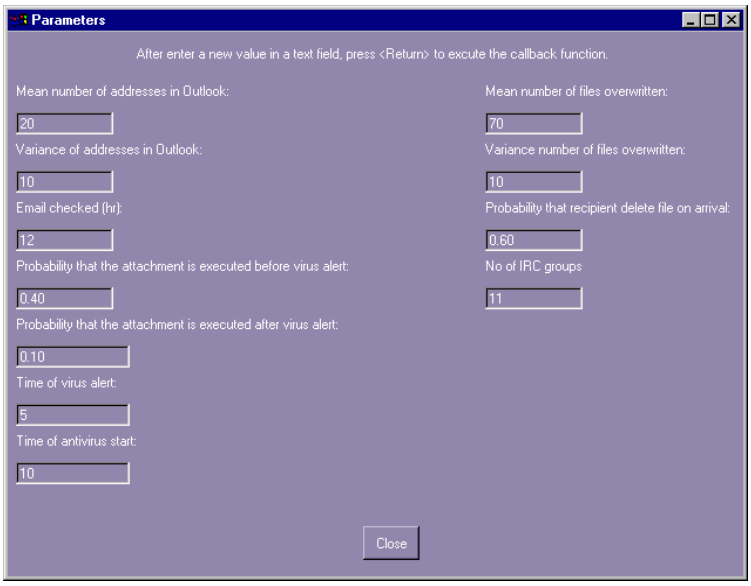


Figure 24. Screen shot of simulation interface for parameter input

Figure 24 shows the parameter input window. The distributional aspects of user specified parameters such as number of addresses in the Outlook address book, generation time (email checked (hr)), and number of files overwritten, are specified by mean and variance. These parameters are used to generate the random states shown in the simulation.

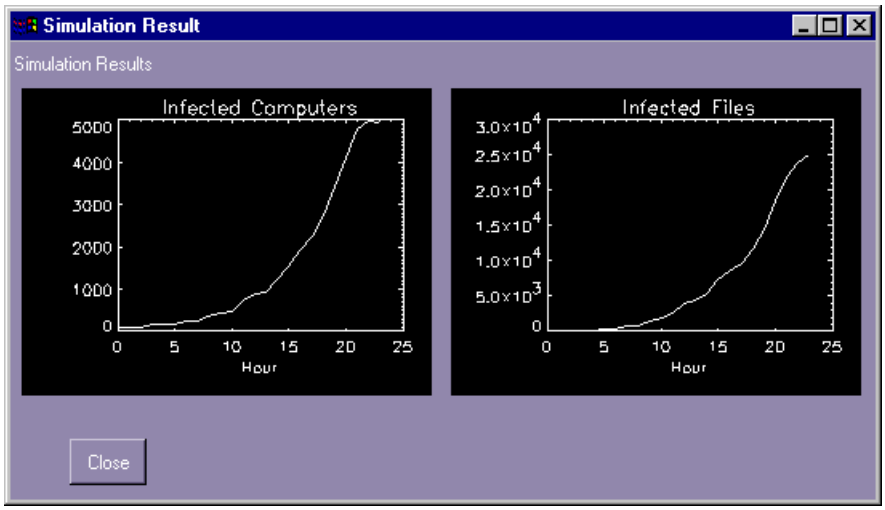


Figure 25. Simulation output

The simulation produces statistical and graphical output. A screen shot of the graphical output is shown in Figure 25.

4.11 Notional concept of operations

4.11.1 Student cyber security education

Some of the transmission parameters identified in this work are anthropogenic. For example, the parameter, *probability of executing an email attachment*, may lend itself to modification by educational efforts similar to health education campaigns. Such a campaign might be targeted at youth, who are often the primary users of computers at school and in the home. Many secondary schools have computer labs and competent students functioning in a limited role as network administrators. An intervention could be delivered during the school year. Industry mentors could contribute immensely in building the practical competence of these students in dealing with security threats for the school, their families, and other students. The effectiveness of the intervention could be monitored annually by written questionnaires or in an ongoing manner (Jorgensen 2002).

4.12 Conclusions

Metrics are needed that succinctly capture the magnitude of cyber threat. BRR provides a basis for comparison based on the potential for infectious spread. This can serve as one basis for resource allocation, particularly when resources must be distributed to combat many threats simultaneously.

Decomposing the phenomenon of the rapid spread of computer viruses and worms in terms of public health analyses allows us to examine different facets of control. Two metrics that may contribute to more effective control strategies are the estimated basic reproduction rate and generation time of a cyber virus or worm. These metrics contain parameters inherent to users and changes in user behavior can be encouraged and their effects measured.

The basic reproduction rate is a linear function composed of the product of the mean number of contacts and probability of transmission. Control strategies that reduce the effective size of the address book, i.e., the number of vulnerable addressees, or reduce the probability of transmission, will lead to reductions in the basic reproduction rate. Reductions may be achieved by scanning email upon arrival, the use of filters and wrappers and the use of non-Windows or non-Outlook software.

The work presented assumes homogeneous mixing in a susceptible population. Barabasi and Albert (1999) demonstrated that large networks such as the World Wide Web are organized as scale-free networks, in which heterogeneity arises from a few highly connected nodes. Recent work by Pastor-Satorras and Vespignani (2001) has examined the spread of computer viruses in scale-free networks. Pastor-Satorras and Vespignani found that in this model epidemics spread relatively slowly and non-exponentially in their early phases. Lloyd and May (2001) note that these scale-free models might be poor models for human interactions, since heterogeneity is usually low in networks describing relationships among individuals. These models may not be the most appropriate ones to model diseases passed by social contact, such as email viruses spread by Outlook address books. From a more practical point of view, in smaller, local networks, homogeneous mixing assumptions may be entirely appropriate, making the simple BRR the tool of choice for predicting the spread of an infectious virus in this more limited domain. Furthermore, if the virus or worm contains a mechanism for limiting spread, it will not fully exploit the scale-free nature of very large networks, and might be

adequately described by homogeneous mixing. For example, Code Red V1 limited its spread to eight possible neighboring IP addresses, in essence fixing the size of its pool of potential hosts.

The focus of this report will now turn to an examination of the broader community, exploring the complex system dynamics of computer networks in Chapter 5.

4.13 References

Bailey NTJ. 1982. *The Biomathematics of Malaria*. Charles Griffin & Company, Bath.

Barabasi AL and Albert R. 1999. Emergence of scaling in random networks. *Science* 286: 509-512.

Halloran ME. 1998. Concepts of infectious disease epidemiology. In: KJ Rothman and S Greenland, eds. *Modern Epidemiology*. Lippincott-Raven, Philadelphia.

Hethcote HW. 2000. The mathematics of infectious diseases. *SIAM Review* 42 (4): 599-653.

Jorgensen J. 2002. A public health education approach to security education. Prodeedings of the *6th National Colloquium for Information Systems Security Education (NCISSE '02),* Bellevue WA, 5-7 June 2002. In press.

Kelsey D. May 09 00. Love Bug losses estimated at $6.7 bil, still mounting. http://www.info-sec.com/viruses/00/viruses_050900a_j.shtml

Kephart JO, Chess DM, White SR. 1993. Computers and epidemiology. *IEEE Spectrum*. http://www.research.ibm.com/antivirus/SciPapers/Kephart/Spectrum/Spectrum.html

Kephart JO, White SR. 1991. Directed-graph epidemiological models of computer viruses. *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA May 20-22, pp 343-359. http://www.research.ibm.com/antivirus/SciPapers/Kephart/VIRIEEE/virieee.gopher.html

Kephart JO, White SR. 1992. How prevalent are computer viruses? http://www.research.ibm.com/antivirus/SciPapers/Kephart/DPMA92/dpma92.html

Kephart JO, White SR. 1999. Measuring and modeling computer virus prevalence. *Proceedings of the 1999 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA May 24-25, pp 2-14. http://www.research.ibm.com/antivirus/SciPapers/Kephart/PREV/prevalene.gopher.html

Leyden J. Feb 02 2001. Anna Kournikova virus spreading like wildfire. http://www.theregister.co.uk/content/8/16846.html.

Lloyd AL and RM May. 2001. How viruses spread among computers and people. *Science* 292 (5520): 1316.

Molineaux L, Dietz K, Thomas A. 1978. Further epidemiological evaluation of a malaria model. *Bulletin of the World Health Organization* 56 (4): 565-571.

Pastor-Satorras R, Vespignani A. 2001. Epidemic spreading in scale-free networks. *Phys Rev Lett* 86:3200.

Rosenberg R, Andre RG, Somchit L. 1990. Highly efficient dry season transmission of malaria in Thailand. *Transaction s of the Royal Society of Tropical Medicine and Hygiene*, 84: 22-28.

Sullivan B. May 26 2000. Beware the mass-marketing 'kak' virus. *ZDNet News* http://www.zdnet.com/zdnn/stories/bursts/0,7407,2578195,00.html

White SR, Kephart JO, Chess DM. 1995. Computer viruses: a global prospective. *Proceedings of the Fifth Virus Bulletin International Conference*, Boston, September 20-22, 1995.
http://www.research.ibm.com/antivirus/SciPapers/White/VB95?vb95.distrib.html

5 Cyber Ecology and System Health

In cyber systems as well as ecosystems, there is a general lack of understanding about the workings of the system. Historically, individual ecosystem components have been extensively studied, while the internal structural interactions among components have remained less understood. In computer networks, this pattern holds as well. Individual components designed and manufactured by humans are well studied and understood, although the structure that emerges when these components are assembled into a system with complex interactions is not. In this chapter of the report, we address analytical methods for the high-level analysis of complex cyber systems.

5.1 System health

Health represents a tradeoff among the qualities of performance, vigor, and resilience. When assessing the health of the system we must also consider all of the attributes in this triad. Although one system may be more vigorous in the sense of stability in the face of disturbance, other factors, such as efficiency and recovery are also important when deciding upon a desirable configuration.

Nielsen (1999) has described health in terms of the ecosystem's "capacity for maintaining biological and social organization on the one hand and the ability to achieve reasonable and sustainable human goals on the other." He noted that, "there are two dimensions to the idea of health. The first is the capacity for maintaining organization or renewal, and the second is the capacity for achieving reasonable human goals or meeting needs." The capacity to maintain organization incorporates the ideas of resilience, vigor, and homeostasis. These three attributes contribute the integrity of the system.

In information assurance, we are concerned with analogous attributes of the network:

Ecosystem health	Information Assurance
Vigor; energy throughput	Ability to process information and produce desired results, performance
Organization; complexity; ability to maintain system structure through stress	Ability to maintain system structure and functionality
Resilience; capacity to bounce back when stress is relaxed	Ability to rebound following perturbation; recovery

Various indices are used to assess ecosystem health, such as diversity. Diversity, however, should not be used as a surrogate measure of complexity. Some systems are inherently less diverse than others, but are no less healthy. The arctic tundra, for example, contains far fewer species than a tropical rain forest. Yet within the given environmental constraints, both may be healthy in that they process energy efficiently, they are stable, and they are resilient.

Changes in indices, however, can be used as warning flags. Rapport (1997) lists several ecosystem transformations that signal an ecosystem in distress. These are:

- Reduced vigor – reduction in the energy throughput and level of activity;
- Reduced resilience – reduction in the capacity of the system to rebound following stress;

- Reduced organization – reduction in the level of complexity (e.g., number of species and their degrees of interaction);
- Reduced services – impairment of services due to stress;
- Reduced management options – reduction in uses that might be implemented in unstressed conditions (e.g., recreation);
- Increased subsidies – diversion of resources from other areas to maintain production levels;
- Damage to neighboring systems – exports from one system cause damage to another, either by direct application or competition.

These are the changes that tell us when the system is in distress. We discuss each of these in the context of network health.

- Vigor (Performance): Vigor is a reduction in the throughput of a system. In a database application, throughput is a measure of the total number of queries handled by the server during a given time. A reduction in the number of inputs progressing through the transaction buffer and into storage is a reduction in the vigor, or performance, of the system. In a denial of service attack, packets that are dropped because of the saturation of the capacity of the system to process them are indicative of a reduction in vigor. At a higher level, a reduction in mission throughput because of work interruptions is also indicative of a reduction in the vigor of the system.
- Resilience (Recovery): Networks may exhibit protracted periods for recovery from certain types of attacks. System administrators are often evaluated by the frequency of downtime incidents and their durations. Increases in these factors are indicative of decreased system resilience.
- Organization (Functionality): Network functionality is often tied to specific platforms (ISS, Apache, Novell, to name a small subset). When critical functionality is performed by one specific platform, compromise of that platform places the entire network at risk of failure. Increased diversity within functions providing critical functionality provides redundancy that may be important as the network responds to attack. Reductions in diversity signal potential increased vulnerability. Introductions of such vulnerabilities are often associated with reduced management options.
- Reduced management options: Certain management policies may require that specific services and activities be curtailed. For example, exclusive implementation of proprietary protocols (i.e., Microsoft-only networking services) result in decreased diversity and organizational richness. These are harbingers of potential system distress. Other examples include operating system upgrades that render older versions of certain software applications nonfunctional, effectively removing them from the network ecosystem and implementation of Active-X controls on collaborative web-based clients. Active-X controls constrain the use of browser to Microsoft Explorer and effectively remove other browsers from the local network ecosystem, with a concomitant reduction in network diversity.
- Reduced services: Ecosystem services are natural processes that benefit the users of the network. They are not necessary for the continuation of the network per-se, but they do allow a more robust response to transient insults. For computer networks, examples of ecosystem services include installation of security measures, such as anti-virus software, and system utilities. These services increase the stability of the network and enhance the users' ability to perform work. Decreases in the provision

of such services, for example, due to decreased availability of system and network administrators, are indicative of system distress.

- Increased subsidies: An ecosystem requires subsidies when its internal processes are insufficient to sustain itself. For example, obligatory application of fertilizers to soil indicate that the ecosystem is not functioning at a sustainable level. Likewise, for a computer network, when the system is not sustainable without continuous input of external resources, its overall health should be questioned. Such a situation may occur, for example, when outside consultants are continuously required for network operations.
- Damage to neighboring systems: This distress indicator occurs when one system or service suffers at the expense of another. Often it is the result of a fixed resource base that is insufficient to maintain all critical functions of the network. For example, a policy may be implemented to update software only, not hardware or a choice may be made to maintain email services, but to sacrifice employee web-access.

The health of the agroecosystem is determined by a balance between human activity and the natural environment. Waltner-Toews and Nielsen (1995) describe a healthy agroecosystem as one that "exhibits a high degree of integrity, operates efficiently, has a strong capacity to respond, and meets the reasonable goals of the shareholders," where

- integrity is the degree to which an agroecosystem maintains its organization or structure,
- efficiency is the degree to which an agroecosystem efficiently processes energy and materials, and
- resilience is the ability to cope in the face of unpredictable stressors, and effectiveness is capability of the agroecosystem to meet the reasonable goals of stakeholders.

These qualities correspond to those used to address the health of computer networks. Efficiency is analogous to performance, integrity to functionality, and resilience to recovery. Effectiveness measures the degree to which the network supports its users. The first three qualities, efficiency (performance), integrity (functionality) and resilience (recovery), refer to the sustaining attributes of the network. When evaluating health from an internal perspective, that is, from the perspective of the network, these qualities contribute to the survival of the network when stressed or attacked. Effectiveness applies to the description of the network from an external perspective. Network managers will be interested in the work that can be supported and extracted from the network as well. The internal and external viewpoints must be balanced when examining the overall health of the network.

The challenge is to translate information about network health into management practice. Walters (1986) considers three ways to structure management as an adaptive process. The first a "trial-and-error" approach, in which early management decisions are made at random and later choices are made from a subset that gives better results. The second is the passive adaptive approach, in which historical data is used to determine a single, best model, and decisions are made assuming this model is correct. The third approach is active, adaptive management, where the data currently available are used to construct a range of models. A management alternative that provides acceptable short-term performance in light of its long-term uncertainty is selected from among these models. The management alternative is implemented as an experiment that will be reassessed and refined over time.

In ecological terms, the health of computer networks does not depend on the implementation of specific network topologies. Health emerges from the structure formed by the interconnections and interactions of humans and machines as they complete specific tasks. In fact, low-level management of attacks based on topology alone can waste resources when control is misapplied to the areas of the network that seem important topologically, but are structurally insignificant.

A 'healthy' system is one that is resistant to cyber attack and is capable of recovery. It is desirable to protect key aspects of the system. They may be shielded by the system so that they resist change when stressed. These attributes can be assessed using a community-oriented approach to ecosystem analysis to assess the emergent properties that characterize the system as a whole.

5.2 Ecological analysis of cyber attack

In previous work, we discussed epidemiologic metrics (basic reproduction rate, generation time) that could be used to assess an ongoing threat. At times, however, an attack may transpire so quickly that there may not be time to respond while it is in progress. For example, a recent description of the Warhol worm (Weaver 2001) has extrapolated the complete infection of the Internet in fifteen minutes.

In this situation, in order to stay ahead of the enemy, it is necessary to project beyond the attack itself to its effects. Given that computer networks are hierarchically organized complex systems, the direct and indirect effects of attack on their underlying structure can be predicted. Using these predictions, nodes that are particularly vulnerable can be identified. Foreknowledge of these vulnerabilities can guide the implementation of defensive measures for nodes with strategic value.

Vulnerabilities may be selected for attack for several reasons. One reason is their accessibility. Accessible vulnerabilities present obvious targets to attackers. More sophisticated attackers will target vulnerabilities based upon their importance. These vulnerabilities will provide attackers with the "biggest bang for the buck," causing maximal damage through the direct effects of the attack and through collateral damage resulting from the indirect effects of the attack. Protection of all vulnerabilities is impossible. As one law enforcement official said regarding defense against terrorism, "[t]here are a virtually infinite number of possibilities and only a finite number of solutions…. So, we have to prioritize our level of activity and preparedness" (King, 2001). Our goal in this report is to develop tools to discern the probable intent of a sophisticated attacker.

In this chapter, we assess the effects of direct attacks on cyber systems. These attacks are modeled as inputs to specific system components. The techniques presented in this report form the foundation of a dynamical systems approach to the identification of vulnerabilities. We explore the application of methods used to analyze complex ecological communities for information assurance. We discuss the theoretical underpinning of the work and show through examples how the fundamental concepts map to cyber ecology. We discuss the elicitation of community structure and ways in which this structure may be exploited for strategic cyber defense.

5.3 Qualitative approach to ecosystem analysis

The qualitative approach to the analysis of ecological communities used in the Cyber Ecology Toolkit uses monotonic data to summarize the dynamics of a set of interacting variables. The mathematical foundations of this approach were developed by Lyapunov in the 1890's and developed further through cybernetic analysis (Gardner and Ashby 1970) and economics (Quirk and Ruppert 1965). They were introduced to ecology in the early 1970's (May 1972, 1973). The matrix representation of the community was developed by Levins (1968). Bender *et al.* (1984) demonstrated the utility of the inverse of the community matrix in deriving predictions about the way that a community will respond to stress.

Ecologists view ecosystems in terms of *stability*. A stable ecosystem will return to its equilibrium state after a disturbance. This tendency allows us to predict the behavior of the system. Stability provides a backdrop for evolution and adaptation. Ecosystems also exhibit other emergent properties, such as resilience, that affect the rate and manner of recovery.

Modern ecosystem analysis has focused on the quantitative analysis of systems. This type of analysis has allowed detailed observation of parts of ecosystems. The analysis of the dynamics of large systems, until recently, has been computationally intractable. Recent technological advances provide sufficient computational power to enable qualitative symbolic analysis of ecosystem dynamics. Qualitative analysis allows us to step back and observe the contribution of system structure to the dynamical behavior of complex systems in a cost effective and efficient manner. Qualitative data are more easily obtained than quantitative data. Hypotheses can be rapidly generated, evaluated and revised to gain knowledge about network structure and response to stress. We examine cyber systems from this qualitative perspective in this report.

Natural systems are dominated by negative feedback. These feedback patterns have allowed these systems to persist and evolve over time. From the viewpoint of systems governed by negative feedback, we adopt the following assumptions:

- cyber systems possess some level of stability (based on their historically observed recovery following attack)
- the persistent threat of attack presents a stable backdrop for adaptation of defenses.

However, human systems are often dominated by positive feedback. These systems require the constant input of resources to persist. Defensive systems, such as the defensive deployment of countermeasures, are governed by positive feedback. We discuss both types of systems in this chapter of the report.

5.4 Elicitation of network community structure

The ecological approach to the analysis of system dynamics begins with specification of the ecological community. Interrelationships among community variables are then defined. The community is evaluated with respect to its stability, and predictions are made regarding the effect of input into specific system variables.

5.4.1 Definition of a community

A community is an ecological system consisting of "two or more components that interact" (Hall and Fagan 1956). It exhibits organizational dependence beyond aggregativity, providing a basis for emergent properties. These emergent properties allow the system to perform in a manner beyond what is possible by its individual parts. A system can be defined by what it is not. Four conditions defining aggregativity have been specified by Wimsatt (1997). The conditions that must be negated to provide evidence of a system are:

- Intersubstitution – parts of the aggregate may be rearranged or interchanged with parts of other aggregates with no effect.
- Size scaling – parts may be added or deleted with no effect.
- Decomposition and reaggregation – parts may be decomposed and rearranged with no effect.
- Linearity – cooperative or inhibitory interactions are not present.

Since a system is more than an aggregate, the failure of any one or more of these conditions provides evidence of the existence of a system.

Collections of variables that exhibit only aggregative qualities cannot achieve emergent behavior and are not systems. Computer networks are systems because they do not satisfy the conditions for aggregativity. They fail the criteria for aggregativity in the following ways:

- Inter-substitution – parts in the system may not be rearranged or interchanged with parts of other systems. For example, computers that have been assigned local IP addresses cannot function outside a gateway.
- Size scaling – Addition or deletion of parts may impact a network. For example, removal of an email server will disable the capability to send and receive email.
- Decomposition – Parts may not be decomposed and rearranged at will. For example, all machines in a network may be disconnected and reconnected in a way that precludes proper functioning.
- Linearity – Cooperative and inhibitory interactions are present. For example, two applications using OLE may cooperate in processing information. On the other hand, two applications may also compete for CPU time when running simultaneously.
- The failure of computer networks to satisfy the criteria for aggregativity provides a basis for exploring their system properties. One such property is hierarchy. Ecosystems possess many forms of hierarchy (O'Neil *et al.* 1986).
- Level of organization. This is the most explicit form of hierarchy, where systems are arranged in order of level of organization (e.g., cell, organism, population, community).
- Vertical hierarchy based on rates. Generally, the higher the organizational level, the slower its operating rate. For example, an individual tree responds to changes in light intensity and $C0_2$ concentration on a moment-to-moment basis. The growth of the tree integrates these changes and occurs at a slower rate. The growth of the forest integrates the growth of the individual trees and occurs at a yet slower rate.
- Horizontal hierarchy incorporating groups of functionally similar components. An ecosystem may be modeled as blocks of functionally similar species, or guilds. It

may also be modeled as a set of subsystems, each possessing its own internal hierarchical organization.

It is important to confine the scope of analysis to a manageable depth. While it may seem attractive to incorporate substantial detail in a community analysis, this detail may not be illustrative, and in fact may obscure more general behavioral trends. There is often great inconsistency between what we can measure and what we want to measure. Clear, precise measurement of small-scale interactions will not inform higher-level management decisions about the community in general. Community level analysis can assess short-term trends that are indicative of a more general system breakdown. It can detect patterns of loss that highlight the potential for loss, but cannot pinpoint exactly where the loss will occur.

This concept of ecosystem as 'organized complexity' will permit us to abstract the structure of communities of interrelated variables to predict the behavior of computer networks as they respond to input in the form of external attacks. Although dynamical behavior is a complex mathematical topic, ecologists have found ways of visually expressing systems to make analysis intuitive and accessible. We will discuss these visualization and analysis techniques later in this report. In the next section, we will discuss the components of the general, dynamical ecosystem model.

5.4.2 Specification of a system

Systems are composed of variables and their interactions. Weiss (1971) described the community as "a complex unit in space and time so constituted that its component subunits, by 'systematic' cooperation, preserve its internal configuration of structure and behavior and tend to restore it after non-destructive disturbances." These disturbances are non-destructive in terms of the community. Many individuals may die or be injured in such a disturbance. It is non-destructive in the sense that the community continues to function. Following the ecological paradigm, we will refer to complex systems of variables and interactions that possess feedback as communities. We model attacks as such non-destructive disturbances.

In ecology, variables are often taken to be the size of populations or species. Although intuitive, a strict definition of species is controversial. Regenmortel (1990) developed a very general definition of virus species as a "replicating lineage that occupies a specific niche". For our investigation of community structure in computer networks, we will assume an even more general taxonomic definition of variables developed by Sneath and Sokal (1973). In their work on numerical taxonomy, they used the operational taxonomic unit (OTU) as the base-level aggregate used in classification. The requirement for an OTU is that it be measurable and it can vary in rank from model to model depending upon the level of study. For example, the OTU might be an individual or group in one study or an average output of a process in another. In cyber ecology, a population-level OTU could be the number of packets transmitted or the CPU cycles utilized by a particular application. A process-level OTU could be the tendency of a thread in an executing process to return a page fault.

The concept of community, incorporating a diverse set of mutually distinguishable OTUs at multiple hierarchical levels, differs from its common usage in computer networks. Kumar *et al.* (1999) defined communities on the world wide web as groups of content-creators sharing a common interest. Using link analysis, they identified web

communities by the density of linkages. Indirect linkages, those web sites referenced by pages that point to common web sites but not to each other, demonstrate the richness of the linkages from which the communities were drawn. However, these content-creators and web sites represent diversity at one level of hierarchy, sharing one type of functional behavior. This type of broad, but shallow organization is referred to by ecologists as a guild, and is but one variable of the community. OTUs within the guild are distinguishable by their unique directed core. The strength of their relationships with their consumers can be measured by in-degree, the number of pages that hyperlink to a page.

5.4.3 Variables

A variable is any component that can be measured and can vary. A variable may be a population (a pride of lions), a process (a hunting pressure), or an abstract quality (market value). The units of measurement may be objective or relative. For the purposes of our analysis, we only require that any two quantities be defined as increasing or decreasing with respect to one another, that is, their rates of change are either of the same sign (negative or positive) or of opposite sign.

Variables can be distinguished from one another based on their taxonomic characters. These are features that distinguish one variable from another (Michener and Sokal 1957). Wang et al (2000) simulated the effect of partial immunization (one, five and ten percent immunized) on the propagation of viruses through computer network. They found that as the level of immunity increased, the probability of epidemic decreased and that immunization was more effective in strict hierarchical, rather than cluster networks. From the standpoint of community, the populations immunized at the one percent level are distinguishable from the populations immunized at five and ten percent. Each one of these populations might be taken as an OTU in a community level analysis. It is relatively simple exercise to identify taxonomic units that differ from one another. Identification of the manner in which they interact is more difficult. True community-level analysis seeks to combine many interconnected taxonomic units in one analysis.

5.4.4 Links and cycles (loops)

A link represents the relationship of one variable to another. It summarizes the sign of the derivative of the first order differential equation describing the direction of effect of one variable to another. The possible relationships between any two variables are:

- Amensal – a variable causes a decrease or diminishment in another;
- Commensal – a variable causes an increase or augmentation in another;
- Predator/prey – an interaction between two variables in which a variable, *variable a* (the consumer or predator), causes a decrease or diminishment in another variable, *variable b* (prey), while *variable b* confers benefit to *variable a*. In an ecological relationship, a predator consumes its prey (loss to prey variable) and derives an increase in reproductive potential in return (benefit to predator). The producer/consumer relationship is one form of a predator/prey relationship. Producers (plants) derive energy from the sun and process this energy into food items for other organisms. Consumers ingest these food items, deriving benefit to themselves while diminishing the plant population. A self-regulated plant population or community may recover. In computer networks, the concept of predator and prey

is counterintuitive. We will assume that the interrelationships exhibiting benefit/loss behaviors are producer/consumer relationships.

- Interference competition – an interaction in which two variables engage in a reciprocal inhibitory relationship. For example, two plants that compete for the same space, where one plant actively precludes the other from occupying that space, are engaged in interference competition. This type of competition is distinguished from resource competition, in which two variables consume a common limiting resource but do not interact directly. In a computer network, an application that requires a specific configuration that is not compatible with another functionally equivalent application, such as Novell and Windows NT server, are engaged in interference competition.
- Mutualism – an interaction in which two variables 'mutually' support each other. In an ecosystem, a mutualistic relationship occurs when one species creates a favorable environment for another, perhaps through providing shade or vegetative cover, and in return another species provides support for the first, perhaps through the return of increased nutrients in decomposing by-products. In a computer network, mutualism occurs when two applications provide input into another. A Microsoft Word document that contains embedded Microsoft Excel spreadsheets is the result of a mutualistic relationship.

The relationships within a community are dynamic. A mutualistic relationship may change into a competitive relationship as stresses are applied to the community.
When consuming a resource, we tend to think in terms of supply and demand instead of in terms of relative effects. In a stable system, there will be sufficient resources (e.g., bandwidth) to support consumers of the resource. Resources and services will be sufficient to meet demand.
Self-loops are drawn as effects that loop back to the same variable. They indicate self-regulation and form the foundation for stability. It is the most basic level of the community hierarchy, and the basis upon which all subsequent relationships are built. Self-regulation is the ability of a variable to replenish itself. Each self-regulated variable is a stable subsystem within itself. It derives its regulation through one of several mechanisms:

- Density dependence - The variable is capable of exerting a regulatory effect upon itself, so that it will engage in or limit unrestricted growth. It is commonly represented with the equation of 'logistic growth'.
- Outside resources - The variable regulated by 'outside' factors, in essence deriving energy from outside the system. In this case, self-regulation is a convenient short-cut to simplify the system down to its components of interest. A stable sub-system may be collapsed into a self-regulatory loop. This allows representation of very complex systems in very simple terms.

The absence of self-regulation on a system variable implies that the variable is completely regulated by other variables in the same system. The hypotheses addressed by this configuration are subtle, and we recommend that the novice modeler assign self-regulation to all variables. This strategy yields a more general model that is less subject to misinterpretation. The default value for self-regulation should be negative. Positive self-regulation should be assigned to a variable if it is subject to unregulated growth or

decline. This may be the result of a "harvest," where a variable is significantly consumed without consideration of replacement.

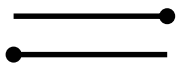
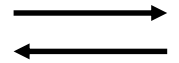
5.4.5 Signed digraphs

Biologists, including ecologists, are visually oriented scientists. For biologists, communities are best represented and analyzed using signed digraphs. These visual models, signed directed graphs, or digraphs, are formed by combining a relatively small number of components.

- Variables are represented as circles: 

- Links are arrows representing positive or negative effects. They can be single or pairwise (Table 9).

Table 9. Link types

Type of pair-wise relationship	Name	Feedback
	Predator-prey	Negative
	Interference	Positive
	Mutualism	Positive
	Commensalism	0
	Amensalism	0

A typical system is shown in Figure 26:

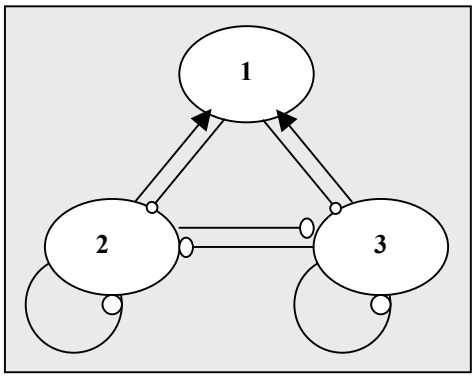


Figure 26. A typical ecological system

Each link represents an element of the community matrix:

$$\begin{vmatrix} Aa1 & +a12 & +a13 \\ 1 & & \\ -a21 & -a22 & -a23 \\ -a31 & -a32 & -a33 \end{vmatrix}$$

Each a_{ij} element represents the effect to variable i from variable j. For example, since there is no self-regulation for variable 1, the a11 element, the effect to variable 1 from variable 1 is zero. Likewise, in the signed digraph is shown as a line ending in a bubble, or a negative effect. The a21 element of the community matrix, the effect to variable 2 from variable 1, is therefore negative.

The community digraph represents the state of the network before it is perturbed. Householder *et al.* (2001) have noted "to be able to detect anomalous behavior, you first must be able to characterize what 'normal' means in the context of your network." The community digraph captures the state of the network in the absence of stress.

Construction of the community digraph is an exercise that illustrates the importance of the environment on the network at large. As we attempt to describe causal explanations about how a network behaves, we may find it necessary to incorporate variables outside of the "traditional" boundaries of the network. The system level approach developed in this report allows the analyst to develop a scenario for a small focused situation and then to place it in the context of a larger one. This approach is different from many IA approaches that seek to defend networks using highly developed, specialized mechanical and computational methods. The goal of IA Cyber Ecology is to develop testable hypotheses about the behavior of the network as a whole and to use the results of such analyses to guide the strategic management and defense of such networks.

5.4.6 Feedback

Feedback occurs when variables interact. The effects of one variable upon another, that is, in turn linked to other variables, cause a cascade of effects through the system. Feedback occurs when these effects fold back onto each other, forming cycles of effects. There are two types of feedback. Negative feedback occurs when the return (feedback) signal is in the opposite direct of the input signal. Negative feedback is thermostatic and can regulate a system by counteracting input. In the signed digraph, feedback between variables 1 and 2 is obtained by multiplying the signs of the links (+ * -) and is negative. The feedback between variables 2 and 3 is positive. Positive feedback will return a signal of the same direction as input; positive feedback is highly destabilizing, although not necessarily undesirable in all conditions.

Mathematically, feedback also occurs at different 'levels', equal to the number of variables and determined by loops of specific lengths. The number of levels of feedback in the system is determined by the number of variables in the system. Feedback at each level is captured by one coefficient of the characteristic polynomial. For example, level one feedback is the coefficient of the λ^2 term. The feedback at each level for the system shown in Figure 26 is:

Level 0: -1
Level 1: -(a22+a33)
Level 2: -(a21a12 +a31a13 –a23a32)
Level 3: - (+a22a13a31 + a33a12a21 - a12a23a31 - a31a23a12)

Negative and positive feedback systems

Feedback forms the basis for predictability in a complex dynamic system. A system, in which progressively higher levels of hierarchy exhibit decreasing levels of negative feedback, is stable. System behavior in stable communities can be predicted. Ecological analysis, in general, has focused on the role of negative feedback in creating stable communities. This is the general pattern in nature. Systems dominated by positive feedback are not stable and therefore do not persist over long periods of time. Positive feedback systems do possess many desirable attributes, however. While a negative feedback system will store capital to sustain itself in times of stress, a positive feedback system will immediately expend the resources it takes in, translating them into growth. The effects of change in positive feedback systems can also be predicted, but on a short-term basis, since the survival of such a system is not assured by its structure.

The paradox of stability is that a stable network is very good at standing still. In order to grow as a whole, a system must be able to assimilate positive feedback and to translate it into growth. However, as long as the system is dominated by low-level positive feedback cycles, it is very vulnerable. A very minor disturbance can send it into a tailspin.

We offer the following guidelines for discerning positive and negative feedback systems. Positive feedback systems:

- Contain at least one positive self-effect. This means that at least one variable behaves in a self-enhancing manner; and
- Receive a constant input of resources to sustain growth. The significant insertion of resources is a requirement for the survival of the system because the system cannot draw sufficient capital using its inherent extraction processes.

Many human-designed systems tend toward positive feedback because their intent is to benefit from an increasing resource. When faced with a decreasing level of resource, the system will collapse and disappear. By definition, such a system would also disappear even in the presence of a constant (as opposed to an increasing) level of input. For example, agroecosystems that are managed intensively to provide food and fiber for human populations are often not stable. They require the application of allochthonous resources, such as petrochemical fertilizers and are dominated by positive feedback. Similarly, cyber systems that require the continuous input of resources, such as those required to maintain defensive countermeasures, are dominated by positive feedback. These systems are discussed in section 5.7 of this report.

Negative feedback systems are discussed in section 5.6 of this report. These systems:

- Contain at least one negative self-effect. This low-level negative feedback provides a mechanism for the system to conserve resources.
- Are self-sustaining. They can persist solely on resources derived from the environment using extraction processes contained in the system. While such systems are better able to withstand an interruption in resource flow, they do not grow.

Many systems will consist of a mixture of positive and negative low-level feedback, that is, a mixture of positive and negative self-effect loops. When such a system grows in

response to a constant influx of external capital, we say it is dominated by positive feedback or is a positive feedback system. These systems are not stable and will collapse when the insertion of external resources is interrupted. The theory of positive feedback systems has not been fully explored and is not completely understood. When a system is capable of conserving capital extracted from the environment, that is, it possesses strong low-level negative feedback, we say it is dominated by negative feedback, or is a negative feedback system.

Feedback at higher levels also impacts system behavior. Stable systems possess negative feedback at low levels. High-level positive feedback loops, that is, positive feedback loops that involve many variables, are also potentially destabilizing. These long loops are generally introduced when systems are managed. They introduce time delays so that the management of a process may take longer than those in the unmanaged system. When the variables in a system are unable to adjust, the system becomes unstable. Complex systems may be hierarchically constructed so that the variables are actually small systems in themselves. When high-level positive feedback is present that allows the system to grow, these smaller subsystems that are dominated by negative feedback, can provide protection against the total collapse. When the level of resources available to the system decreases so that growth cannot be maintained, the system structure may break apart. However, subsystems with negative feedback will persist and may be 'reconnected' when the opportunity arises again.

5.4.7 Emergence of the graph as a succinct representation of mechanism and causality

The evaluation of complex dynamic systems has benefited from recent advances in graphical analysis. Judea Pearl (1997) has referred to the development of graphs in modeling causal relationships as a fundamental advance of the past decade. In ecology, in particular, graphical representations of complex ecosystems have made it possible for non-mathematicians to undertake mathematical analyses of these systems. Biologists, including ecologists, rely heavily on visual analysis in their respective domains, and the ability to depict complicated systems as a collection of interconnected nodes and arcs provides a visually accessible and intuitive way to summarize a wealth of information. In the structural approach to the analysis of complex dynamic systems undertaken in this project, graphs provide a "fundamental notational system for concepts and relationships that are not easily expressed in any mathematical language (e.g., equations or probabilities) other than graphs" (Pearl 1997). They fill a linguistic gap and provide a convenient shorthand for expressing the manner in which variables in these systems are interconnected. Equally important are absent interconnections that indicate lack of direct interaction between variables.

5.4.8 Model structure

In this section, we review some basic modeling considerations. The model components described in this section form the basis of more complex performance-based and assurance-based models discussed in later sections of this report.

The fundamental relationship in the ecological models is the producer-consumer relationship, where one variable produces input for another with concomitant negative feedback. This relationship can be described in information systems as a transformation

of information. The transformative process through which raw data is converted into useful information was described in the context of the information battlespace by Alberts *et al.* (1999):

> "The key to understanding the roles of and the relationships among battlespace entities is to focus on processes that turn raw data into information, and information into knowledge…. Data are individual facts, measurements, or observations which may or may not be sufficient to make a particular decision. Information is obtained when elements of data are assembled, reconciled, fused, and placed in an operational context. Knowledge is derived from being able to use information to construct and use an explanatory model based upon an understanding of the situation or phenomenon. Such a model allows us to forecast future states, predict outcomes, and also contributes to our ability to control the situation – or to be proactive rather than reactive. This is, of course, a primary goal of command and control."

In the ecosystem approach to understanding system dynamics, indirect effects are expressed through feedback. Transmission along feedback loops allows a system's response to input to be distributed throughout the system. The producer-consumer relationship is a simple feedback loop. Consider two variables: *useful data* and *information*. As more *useful data* are produced, more *information* can be assembled. The link from data to information is positive because they either both increase or both decrease. Feedback is formed by the returning effect from *information* to *useful data*. As the amount of *information* increases, the amount of *useful data* decreases because more of the data will be redundant. In the feedback relationship, the variables are linked in an inverse relationship that is summarized by a negative link. The signed digraph for these two variables is shown in Figure 27.

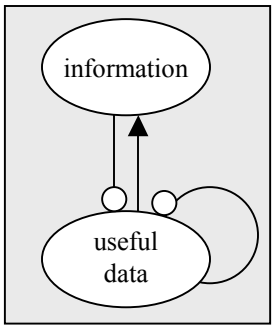


Figure 27. *Useful data/information* producer/consumer relationship

The same pattern of producer and consumer is evident in other models. Consider the relationship between *available bandwidth* and *IIS*[8] *performance* in the performance-based model shown in Figure 28. The relationship to *IIS performance* from *available*

[8] Internet Information Services, a Microsoft Windows-based Web server

bandwidth is positive. This means that as the amount of *available bandwidth* increases, the *IIS performance* improves (within some bound; the relationship is not infinite). Conversely, decreased *available bandwidth* reduces the quality of *IIS performance*. Since the relationships are linked in this manner, the relationship to quality of *IIS performance* from *available bandwidth* is represented by a positive link. In the feedback relationship, improved *IIS performance* reduces the amount of *available bandwidth*, represented by a negative link.

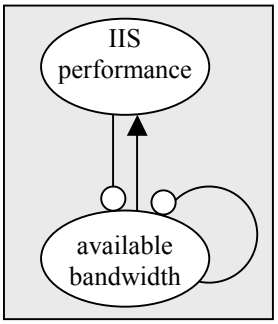


Figure 28. ***Available bandwidth/IIS performance*** **producer/consumer relationship**

The pattern is repeated in the assurance-based model in the relationship between *available sysadmin time* and *level of assurance* shown in Figure 29. As more system administrator time becomes available, the *level of assurance* increases. As less system administrator time is available the *level of assurance* decreases. The variables are linked in tandem as they rise and fall and the relationship to *level of assurance* from *available sysadmin time* is represented by a positive link. In the feedback relationship, as the *level of assurance* decreases, *available sysadmin time* decreases, since more time is devoted to maintaining the high level of assurance. This inverse relationship is represented by a negative link.

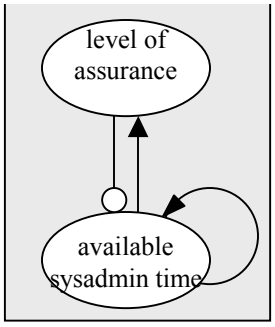


Figure 29. ***Available sysadmin time/level of assurance*** **producer/consumer relationship**

In addition to the producer-consumer relationship, other relationships occur commonly in ecosystems. Reciprocal relationships may also be competitive or mutualistic. In a competitive relationship, variables are linked to each other with negative links. They are inversely related so that an increase in one variable will cause a decrease in the level of the variable to which it is linked. In a mutualistic relationship, variables enhance each other so that an increase in one causes an increase in another. Relationships need not be reciprocal and one-way arrows are permissible.

In Figures 27 and 28, a negative self-loop is shown on the producer variable. This loop indicates that the growth of the producer variable is internally or self-regulated. The producer variable exists in a limited quantity and its growth is density dependent. The response behaves in a thermostatic manner. When the level is low, more resource is made available to achieve a more or less constant level in the system, and vice-versa. In the *useful data/information model*, the amount of useful data grows quickly when there are few data and more slowly when there are many. (Although the total amount of data may continue to increase without bound, the amount of useful data, that is, data that can be used to synthesize new information, does not). In the *available bandwidth/IIS performance* model, the amount of *available bandwidth* increases quickly when the pipeline is empty and more slowly when there is heavy utilization of bandwidth.
In addition to negative self-effect, positive self-effects also occur in the models. In the *available sysadmin time/level of assurance* model shown in Figure 29, a positive self-loop on *available sysadmin time* is shown. This link indicates that the amount of *available sysadmin time* present in the system is 'harvested' (i.e., removed at a rate independent of community feedback). As *available sysadmin time* is consumed, the variable becomes less likely to be able to restore itself. For example, consider system administrator activity during a viral outbreak. As *available sysadmin time* is devoted to combating the outbreak, less time is available to engage in maintenance activities that optimize system administrator productivity, creating further problems that consume more system administrator time. The effect is self-amplifying. The positive self-effect loop captures this vicious cycle.

5.4.9 Stability criteria

Stability is the ability to recover from a disturbance. Mathematical criteria are used to assess system stability. In this report, we describe criteria that can be used to address qualitative stability using monotonic data (data that are either continuously increasing or decreasing). In other words, we only need to know the direction of the effect from one variable to another (i.e., the sign of the derivative). By assembling these data into a community matrix and applying the Routh-Hurwitz criteria for stability, we can determine whether this qualitatively specified system will return to its pre-disturbance equilibrium state following the disturbance.
Eigenvalues can be used to describe system behavior (Jorgensen 2000). A system is stable if all its eigenvalues (roots of the characteristic polynomial) have a negative real component. Eigenvalues cannot always be evaluated directly, and the coefficients of the characteristic polynomial can be used to determine if they can exist. There are two traditional criteria for stability, often called the Routh-Hurwitz criteria. Both have intuitive interpretations:

- all coefficients of characteristic polynomial must be of the same sign. The coefficients represent the different levels of feedback mentioned above.
- Hurwitz determinants must all be positive. The Hurwitz determinants are formed using the coefficients of the characteristic polynomial. This operation determines the relationships of feedback levels to each other, ensuring that lower levels of feedback are stronger than upper levels.

5.4.10 Eigenvalue structure

The behavior of the community following perturbation is expressed in its eigenvalues. Eigenvalues may contain real and complex parts, indicating whether or not a community will recover from perturbation and if it will, whether or not the return will follow an asymptotic or oscillatory path.

Real Part	Imaginary Part	Behavior
<0	0	stable, asymptotic recovery
<0	not 0	stable, damped oscillatory recovery
0	0	neutral stability, displacement from equilibrium persists
0	not 0	neutral, permanent oscillations
>0	0	unstable, movement in direction of disturbance
>0	not 0	unstable, undamped increasing oscillations

In general, hierarchical structures composed of straight chains exhibit the highest level of oscillatory behavior as they recover. This behavior creates a vulnerability, since a second perturbation coinciding with a trough in the oscillatory cycle can disturb the system to such an extent that recovery is impossible.
We can express the system shown in Figure 26 qualitatively, in matrix form.

$$\begin{vmatrix} 0 & 1 & 1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

This matrix quantitatively describes the structure shown in Figure 26. The symbolic quantities have been replaced with 1 (for positive values) and –1 (for negative values). Absent links are represented by the value, 0.
The characteristic polynomial for this system is $-\lambda^3-2\lambda^2-2\lambda$. Its solutions, or eigenvalues, are 0, -1+i, and -1-i. This system will return to equilibrium in a damped oscillatory manner.

While the structure of simple systems is easily elucidated, larger systems are more problematic. As systems become more complex, their stability becomes dependent the growing number of interconnections within the system. The conditions under which stability can occur become more numerous, but less likely. We operate under the assumption that some stability is possible in any system. Instead of determining whether or not the system is stable, we evaluate tendency of the system to behave in a stable manner. A system will tend to respond in a stable manner if feedback at all levels is negative and if feedback at low levels is stronger than feedback at higher levels.

5.5 Exploitation of network community structure

5.5.1 Qualitative expression of a community

Measurement of interactions in any community is problematic. Aside from the sheer volume of observations that must be collected and processed, it is not always clear that the quality measured captures the property of interest. Often, however, analysts possess a general idea of the properties of interest and can elucidate the direction of effect of one variable with respect to another. For example, although the rate of decrease in failed messages during a DDoS attack may be difficult to measure, the fact that a decrease has occurred is often readily apparent. These monotonic relationships that represent the direction of change of one variable with respect to another are sufficient to elucidate the underlying structure of a community.

5.5.2 Prediction

When a complex system (community) possesses a local stable equilibrium point, it will return to this level following a minor perturbation (homeostasis). This provides us with a basis on which to make predictions about community behavior. The stability assumption may be relaxed to the assumption that the community possesses a stable trajectory (homeorhesis). Under this assumption, the relative magnitudes of the variables and their interactions are sufficient for prediction.

Predictions are derived from the inverse of the negative community matrix. The adjoint (the product of the inverse and determinant) of the matrix provides the same qualitative information, assuming that the first Routh-Hurwitz criterion is satisfied (i.e. negative feedback at all levels). By convention, a permanent input (or *press*) is positive (the signs of the elements of the adjoint are reversed for negative change) and the indirect effects (elements of the adjoint or inverse) are read down the column of the site of input. For the above matrix, the predictions are:

(Input into variable 1)		
-	0	0
0	+	-
0	-	+

The counterintuitive nature of the results can be seen from the top left element. In this case, a positive input to variable (1) will result in its decrease and have potentially little impact on the rest of the system. The zeros represent canceling cycles, and in actuality change may depend on actual values. However, it is a rigorous conclusion that, for this input, there will be little significant impact other than at the site of input (1).

The prediction matrix shows the results of stress as an input to one variable as it is manifested throughout the system as a whole. Predictions are valuable because they elucidate the indirect targets of an attack. An attack on one specific variable may merely be a steppingstone to a further objective. As attackers become more sophisticated, they will acquire the knowledge and skills necessary to exploit system structure to achieve a desired disruption.

A system dominated by positive feedback will not possess a stable equilibrium. Nonetheless, the predictions may still apply. Predicting the effect of positive feedback simply requires a reversal of normal predictions. This fact arises from the mathematical consideration that positive feedback is expressed as a positive determinant, which is the denominator of predictions. By convention, this is generally assumed to be negative (i.e., 'stable'); a positive feedback simply means a positive determinant.

5.5.3 Purposeful perturbation of the community to obtain information about structure

Ecological experiments elucidate structure by observing community response following perturbation, a selective alteration of the density of one or more members of the community. There are two types of perturbations: *pulse* and *press*. A pulse perturbation is a more or less instantaneous applied to a variable. Community response is observed as the community relaxes back to its equilibrium state. A press perturbation is a more sustained alteration. One or more variables may be modified or eliminated and the system is observed as the unperturbed variables achieve a new equilibrium state. Pulse experiments yield information about direct interactions. The information produced by press experiments results from the combination of direct and indirect effects as the effects of the perturbation manifest throughout the community. System recovery following a short duration ping-of-death is fundamentally and structurally different from system response to a prolonged ping-of-death attack to which the community must adapt to survive.

Each element in the prediction matrix is derived from the sum of the numbers of negative and countervailing positive feedback loops. When there are more positive than negative loops, the value is one. When negative loops outnumber positive loops the value in the prediction matrix is –1. When the numbers of positive and negative loops are perfectly balanced, the value is 0. As systems grow in size, the number of feedback loops becomes very large. A positive value due to an imbalance of one loop becomes far less significant than a positive value due to many more positive than negative loops. Weighted predictions (Dambacher 2001) describe the contribution of structure to each element of the prediction matrix by capturing the strength of imbalance, or ambiguity. Each element in the weighted prediction matrix has been normalized with respect to the total number of feedback loops within the system. A value of one in the weighted prediction matrix means that the prediction is unambiguous structurally; all contributing feedback loops are or the same sign. As values decrease in magnitude, the prediction becomes more ambiguous and is formed by both positive and negative loops, with the sign of the prediction indicating whether positive or negative loops are more numerous.

5.5.4 Analyzing cyber ecosystems using the Cyber Ecology Toolkit

We summarize the steps in constructing and analyzing a signed digraph of a dynamic system using the simulation tool accompanying this report in Table 10.

Table 10. Constructing and analyzing a signed digraph of a dynamic system

Constructing and Analyzing a Signed Digraph of a Dynamic System
Step 1: Select variables Set level of focus. The variables should be roughly of the same order of importance or magnitude relative to the situation being modeled. Analogy: Even though elephants are much bigger than ants, they should both be included in the same model if they contribute important effects. For example, both affect soil friability. To exclude one would not give a complete picture of the soil disturbance.
Step 2: Lay out the variables in a logical order The power of the signed digraph is that it allows the visual tracing of effects through the system. Lay out the variables in a way that makes sense. They can be arranged in the order in which they appear in time or their physical proximity. As understanding increases, the order of the variables may be changed.
Step 3: Set self-effects Negative self-effects form the backbone of a stable system. If a variable does not have a self-effect, this means that its level is completely determined by the inputs feeding into it from other variables. Analogy: In biological systems, the ultimate source of energy is the sun, but the sun is not explicitly included in all models. In models where plants are modeled as the base resource, they are generally assigned a negative self-effect, representing exogenous input from the sun. Similarly, in a cyber system model, bandwidth may be modeled as the base resource for a network. Models with no basal self-regulation cannot be stable. This means that they will persist as long as conditions are favorable and will not recover autonomously to their pre-disturbance state. This does not mean that recovery is impossible. It does mean that recovery must be completely engineered. The system possesses no stable equilibrium.
Step 4: Set connections among variables The connections between variables show whether or not the effects to a variable from another enhance or inhibit growth. Mathematically, the links in the signed digraph represent the partial derivatives of the differential equations describing the dynamic relationships, or links, between the variables. In qualitative analysis, we are concerned with the signs of the links. This allows generation of rapid predictions with minimal data. The digraph allows us to depict these relationships visually and to construct complicated systems using simple rules. Is there feedback? When the effect of a variable's behavior returns to affect its own level through another variable or other variables, feedback is present in the system. Analogy: The fundamental biological feedback loop is the predator-prey relationship. Lions eat gazelles. Since lions reduce the number of gazelles by killing them, the effect to gazelles from lions is negative. However, gazelles provide nutrition for lions so that they can reproduce. The link to lions from gazelles is positive. Thus the feedback from lions, through gazelles back to lions, is negative, the product of the signs of these links. Any relationship in which the product of one variable is depleted and consumed for the benefit of another can be modeled as a predator-prey, or producer-consumer relationship. Other possible feedback relationships between two variables are mutualism and competition. In a mutualistic relationship, two variables mutually enhance each other. In a competitive relationship, two variables compete for resources so that each limits the

other's growth.
Example: System administrators must often maintain multiple operating systems. The systems compete for 'cognitive real estate'. That is, the system administrators cannot focus on one operating system without excluding the others and they demand time and attention synchronously. These operating systems are in competition.
The relationship does not necessarily need to reflect direct feedback. One-way relationships are also common and may be depicted as one-way links in the digraph.
IMPORTANT: Be sure to limit your digraph to first order, or direct, effects. The signed digraph of the community should contain only the direct effects of one variable upon another. Indirect effects that are transmitted through other variables are captured by the mathematical analysis of the system.

Step 5: Derive the community matrix.
The community matrix is a numerical representation of the signed digraph. Each element (a_{ij} -entry) in the matrix is read as the effect to variable *i* from variable *j*. While it is possible to generate this matrix manually, it is not recommended. Experience has shown that this transcription usually contains operator errors. To generate the community matrix using a computer, transcribe the signed digraph into the digraph editor, PowerPlay, and then use the *show matrix* command in the Options menu.

Step 6. Formulate predictions
Predictions reveal the result of increasing or decreasing the level of one variable on the other variables in the system as the system recovers to its pre-perturbation equilibrium state. The predictions show the indirect effects of change on the system. (Recall that the signed digraph described direct effects only). The prediction matrix is the inverse[9] of the community matrix generated in Step 5. The Toolkit software will generate a prediction matrix for you.
To read the prediction matrix for a negative feedback system, read across the columns to determine the site of input. Then read down the column to find the predicted direction of response for a particular variable. By convention, the prediction matrix gives the results, in terms of changes in 'abundance' of positive input into variables only. To read the results of negative input from the table, reverse the signs in the prediction table.
Note that the predictions will be reversed for a positive feedback system. Predictions may be negative ("-" or decrease), positive ("+" or increase) or null ("0" no effect). The effects of multiple inputs may be read from the prediction table only when the inputs are of like sign.

Step 7. Evaluate weights of predictions
Some qualitative predictions are more structurally ambiguous than others. The sum of loops contributing to a response can be positive, negative, or null. Predictions are formed by evaluating subsets of loops in any given system. When all the loops in a subset are all of a given sign, say positive, then we can be reasonably confident that the final prediction will be positive as well. Often, however, predictions are formed by combinations of positive and negative loops. When half the loops contributing to a prediction are positive

[9] The change in species population in a complex system is determined by the matrix equation $\mathbf{N} = \mathbf{A}^{-1}\mathbf{K}$, where **N** is the number of individuals in the population at equilibrium. **K** is the number of individuals that can be supported by the environment, and is presumed to be a constant. When the system is perturbed, elements of **A** change. The effect of this change on the number of individuals is determined by the inverse of the prediction matrix, $-\mathbf{A}^{-1}$ (Ricklefs, 1990)

and the other half are negative, we are less confident about the final direction of the prediction. In this case, the final direction of the effect in the real-world system will be determined not by structure, but by the magnitudes of the interrelationships. The Toolkit contains a facility for evaluating the structural ambiguity of the qualitative predictions. Research using simulations has shown that we can be confident in qualitative predictions of weight 0.5 or greater.
Step 8. Simulate to estimate likelihood of stability Nearly every system possesses some locus of stability. In some, it may be vanishingly small. Others possess a broad plateau of stability. How can we discern among these? The Toolkit contain a simulation facility that will generate 5,000 numerical instantiations of the system, evaluate them for stability, and display the results. Systems that are prone to instability will return few stable simulation models, while those that are more stable will return a greater number.

5.6 Examples of negative feedback systems

We present four examples of negative feedback systems. These systems are potentially stable when negative low-level feedback is sufficiently strong. The examples discussed are:

- Logistics system scenario. This system describes a process contained in a hypothetical logistics system. The underlying mathematics of the analysis are shown in detail. The structure of the system is modified with both increased low-level and high-level feedback to illustrate the resulting changes in stability.
- Propheteer Strawman Scenario. This example contains variables at a higher level of aggregation that those in the previous example. It is a more general analysis designed to show how a community level ecological analysis might be performed to learn about the general tendencies of a large system.
- Code Red. This is LAN-level model that illustrates the effects of resource competition in a negative feedback system.
- CPU-centric model for availability. This is machine-level model designed to show the effects of the release of the Morris worm on the machine-level work.

5.6.1 Logistics system scenario

In this section, we present a detailed community-level ecological analysis. We constructed a model of a hypothetical logistics system and modeled it in the form of the general process shown in Figure 30. We formulated a possible instantiation of this process (Figure 31). We discuss the mathematical equations driving the analysis and then show how the Cyber Ecology Toolkit allows the rapid, comparative analysis of this system and variations.

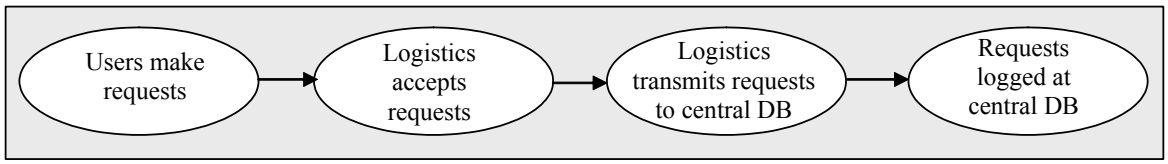


Figure 30. General tasks in the logistics system scenario

The signed digraph of the system depicted linearly in terms of task order is shown in Figure 31. The variables are:

- Number of items in central database;
- Number of requests to central database;
- WAN availability;
- Number of items in the local database;
- Number of logistics orders;
- Number of unfulfilled requests.

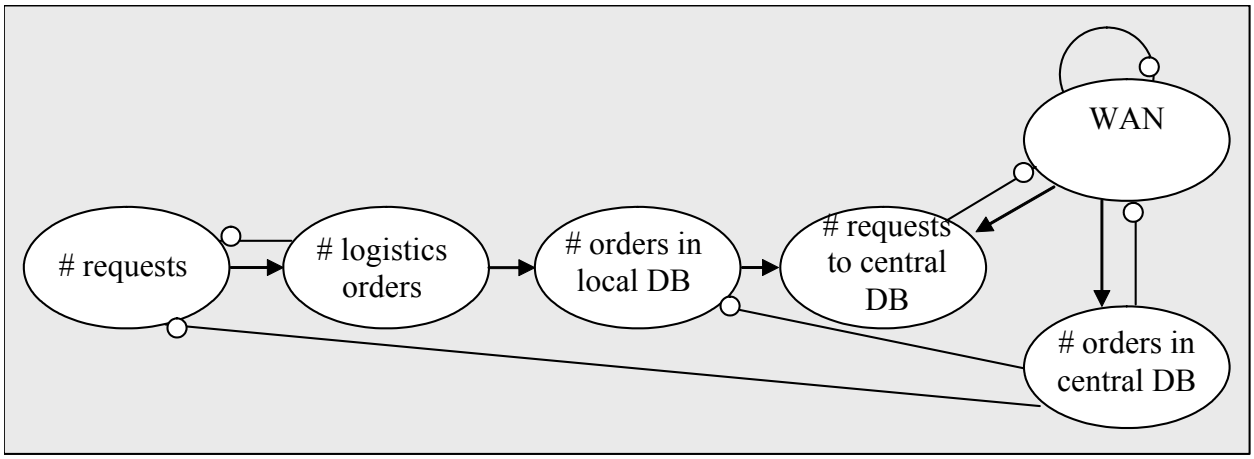


Figure 31. Process-based arrangement of variables in the logistics system scenario

An alternate, more 'biological' ordering of these variables is shown in Figure 32.

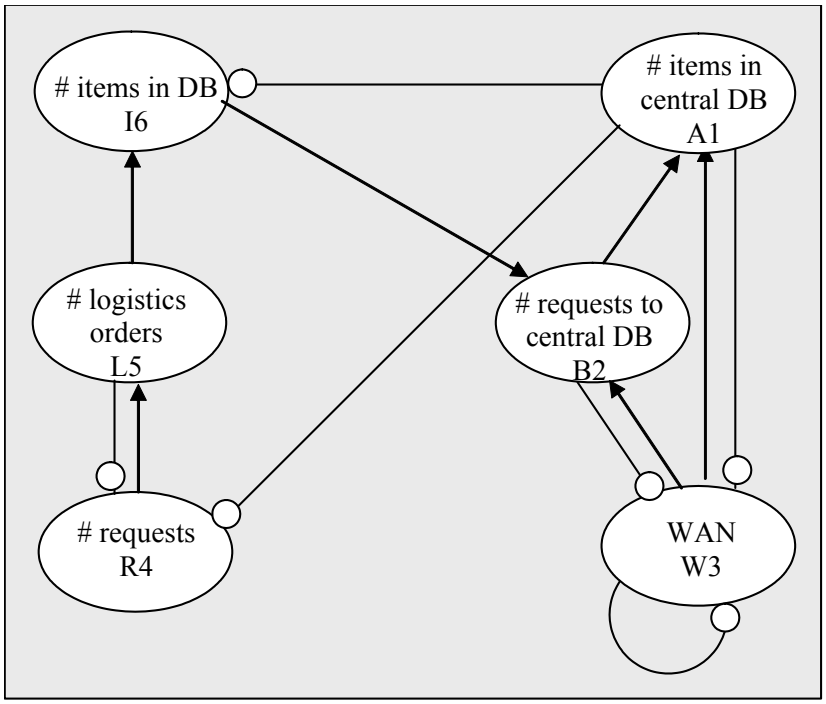


Figure 32. 'Biological' arrangement of variables in the logistics system scenario

The system is relatively simple. The requisitioning process adds a long circuitous path counterbalanced by a short link back to the WAN. The simple two variable system is complicated by a link that passes through four intermittent variables (Figure 33). This means that even though the system seems complex because it contains six variables, it is relatively simple with respect to the number of links.

The only exogenous input into this system is through the WAN. This means that the entire system is affected by the availability of this resource. One consequence of this is that when the availability of WAN is reduced without a concomitant reduction in the remaining variables, the system will 'thrash'. This is directly analogous to application-level thrashing that results when an application is allocated insufficient memory.

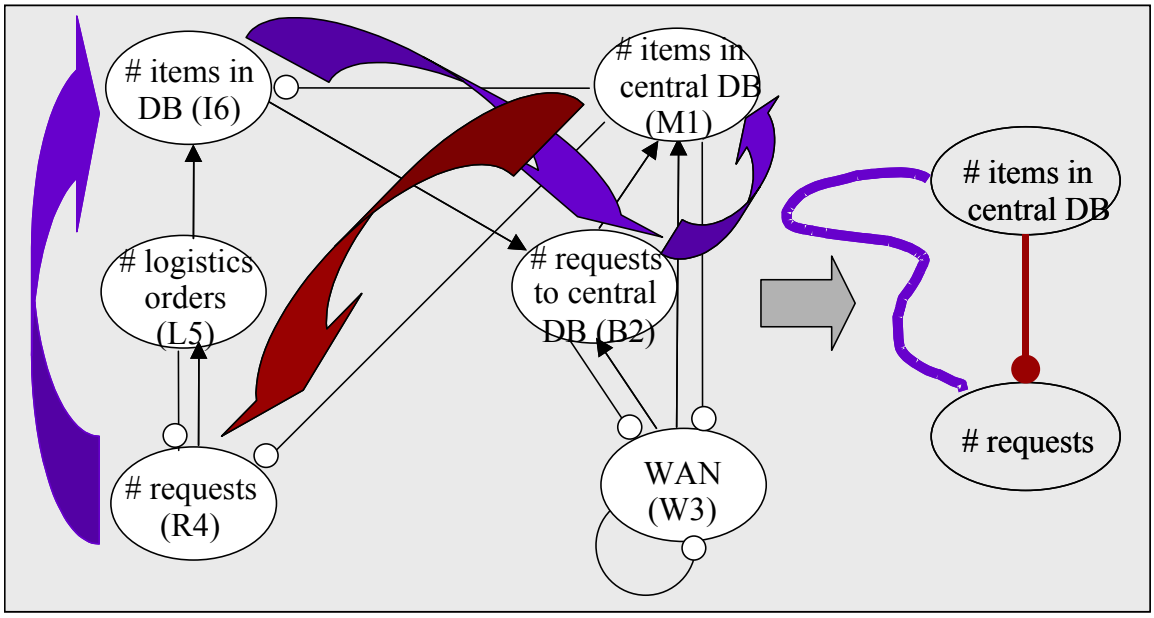


Figure 33. Long and short paths connecting number of requisitions and number of items in the central database

Mathematical Analysis

The work performed by the logistics ecosystem can be represented mathematically by the following sets of differential equations:

Variables are identified by index letter:

- Number of items in central DB A
- Number of requests to central DB B
- WAN availability W
- Number of items in local DB I
- Number of logistics orders L
- Number of requests R

Links identified by index numbers as an interaction term a_{ij} (to i from j).

$$\frac{dA}{dt} = a_{12}AB + a_{13}AW;$$

$$\frac{dB}{dt} = a_{23}BW + a_{26}BI;$$

$$\frac{dW}{dt} = -a_{31}AW - a_{32}BW + W(a_{33} - a_{33}W);$$

$$\frac{dR}{dt} = -a_{41}AR - a_{45}LR;$$

$$\frac{dL}{dt} = a_{54}LI - a_{61}AI.$$

The community matrix, A, is given by the Jacobian:

$$A = \begin{bmatrix} \frac{dA}{\partial A} & \frac{dA}{\partial B} & \frac{dA}{\partial W} & \frac{dA}{\partial R} & \frac{dA}{\partial L} & \frac{dA}{\partial I} \\ \frac{dB}{\partial A} & \frac{dB}{\partial B} & \frac{dB}{\partial W} & \frac{dB}{\partial R} & \frac{dB}{\partial L} & \frac{dB}{\partial I} \\ \frac{dW}{\partial A} & \frac{dW}{\partial B} & \frac{dW}{\partial W} & \frac{dW}{\partial R} & \frac{dW}{\partial L} & \frac{dW}{\partial I} \\ \frac{dR}{\partial A} & \frac{dR}{\partial B} & \frac{dR}{\partial W} & \frac{dR}{\partial R} & \frac{dR}{\partial L} & \frac{dR}{\partial I} \\ \frac{dL}{\partial A} & \frac{dL}{\partial B} & \frac{dL}{\partial W} & \frac{dL}{\partial R} & \frac{dL}{\partial L} & \frac{dL}{\partial I} \\ \frac{dI}{\partial A} & \frac{dI}{\partial B} & \frac{dI}{\partial W} & \frac{dI}{\partial R} & \frac{dI}{\partial L} & \frac{dI}{\partial I} \end{bmatrix}$$

where we write $\frac{dA}{\partial A}$ for $\frac{\partial}{\partial A}\left(\frac{dA}{dt}\right)$.

Completion of this matrix with quantitative data may not be realistically feasible. By the time data are collected, the significance of any analytical findings may have passed. In order to expedite analysis and take advantage of a wider range of data, we populate the

matrix with qualitative data. We assume that the relationships are monotonically increasing or decreasing over the model represented by the matrix. (If they are not and if the model is believed to be stable (i.e., it has persisted for some period of time), we may either have an incomplete system specification or data that are too detailed.) We populate the community matrix with values of 1 (monotonically increasing in value), -1 (monotonically decreasing in value) or 0 (no change in value). These values allow us to compute many attributes of the community quickly and with little data.
The general form of the community matrix from the digraph is:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix}$$

From the community digraph, the qualitatively specified community matrix for the logistics ecosystem is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

To assess the stability of the system, we first take the characteristic polynomial of the community matrix A:

Characteristic polynomial = $\lambda^6 + \lambda^5 + 3\lambda^4 + 3\lambda^3 + 2\lambda^2 + 3\lambda$.

The characteristic polynomial is of the form $a_0\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \ldots + a_{n-1}\lambda + a_n$, where a_0 is always positive.
For the system to be stable, feedback at all levels must be negative and low-level feedback must be greater than high-level feedback. To pass the first criterion, all coefficients of the characteristic polynomial must be of the same sign. The second criterion is assessed by the signs of the Hurwitz determinants. In a stable system, alternate Hurwitz determinants up to order *n* will be positive. The Hurwitz determinants are formed by the coefficients of the characteristic polynomial in the form:

$$H_1 = a_1$$

$$H_2 = \begin{vmatrix} a_1 & a_3 \\ a_0 & a_2 \end{vmatrix}$$

$$H_3 = \begin{vmatrix} a_1 & a_3 & a_5 \\ a_0 & a_2 & a_4 \\ 0 & a_1 & a_3 \end{vmatrix}$$

.
.
.

$$H_n = \begin{vmatrix} a_1 & a_3 & a_5 & a_7 & a_9 & a_{11} & \cdot & a_{2n-1} \\ a_0 & a_2 & a_4 & a_6 & a_8 & a_{10} & \cdot & a_{2n-2} \\ 0 & a_1 & a_3 & a_5 & a_7 & a_9 & \cdot & a_{2n-3} \\ 0 & a_0 & a_2 & a_4 & a_6 & a_8 & \cdot & a_{2n-4} \\ 0 & 0 & a_1 & a_3 & a_5 & a_7 & \cdot & a_{2n-5} \\ 0 & 0 & a_0 & a_2 & a_4 & a_6 & \cdot & a_{2n-6} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_n \end{vmatrix}$$

Usually, complex systems are stable for certain values and unstable for others. Any qualitatively specified system with feedback loops of length greater than 2 will at best be conditionally stable. That is, the only globally stable system of more than two variables is a straight chain system with self-regulation on one or more variables. Conditional stability statements can be assessed by examining the symbolic form of the characteristic polynomial. Conditionally stable system contain more high-level feedback relative to low-level feedback, a characteristic often attributable to over regulation.

In general, systems that pass the first Routh-Hurwitz criterion possess a wide range of stability. Those that pass the first criterion, but fail the second are conditionally stable. In these systems, the relative, quantitative values of the variables will determine the stability of the system. Those that fail both the first and second criterion contain inherent structural flaws that are inconducive to stability.

The Cyber Ecology Toolkit uses simulation to determine the relative stability of a given system. The program randomly generates 5,000 models and tests each model using the Routh-Hurwitz criteria. Output is summarized as the number that is:

- likely to be stable (pass first and second criteria);
- manageable (pass first, fail second criterion);
- likely to be unstable (fail first criterion).

In general, when 50 percent or more of the simulated models are likely to be unstable and the system is known to be a negative feedback system, it should be viewed with extreme caution.

For models with 50 percent or more of simulated models likely to be stable and/or manageable, we can assess the predicted behavior of the system to stress. The predictions for a qualitatively specified system reveal the direction of effect after the system returns to equilibrium. Predictions are obtained from the adjoint of the community matrix A. The adjoint is the matrix formed by the cofactors of A. Each element of the prediction matrix contains the response of species i to a permanent change in the growth rate associated with species j, that is, each column of the adjoint matrix shows the direction of change in each variable given a positive change (increase) in the variable associated with that column.
The effects given by the adjoint matrix are not scaled. In order to determine the relative weight of each prediction relative to the number of feedback loops in the system, we divide the adjoint matrix by the permanent. The permanent matrix gives the absolute feedback associated with each cofactor in the adjoint matrix. The weighted prediction matrix describes the direction and reliability (in terms of structural loops) of input into the variable in the ith column on the variable in the jth row. The weighted prediction matrix for the logistics ecosystem is shown below:

$$weightedpredictions W = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The weighted matrix describes the structural contributions of input into a column variable on the rows representing the members of the community. It summarizes the consequences of both direct and indirect effects. All elements of the weighted prediction matrix are unity. This means that all the relationships except those with values of zero are equally vulnerable in terms of the way the logistics ecosystem is constructed.

Baseline system

The signed digraph for the baseline system is shown in Figure 34.

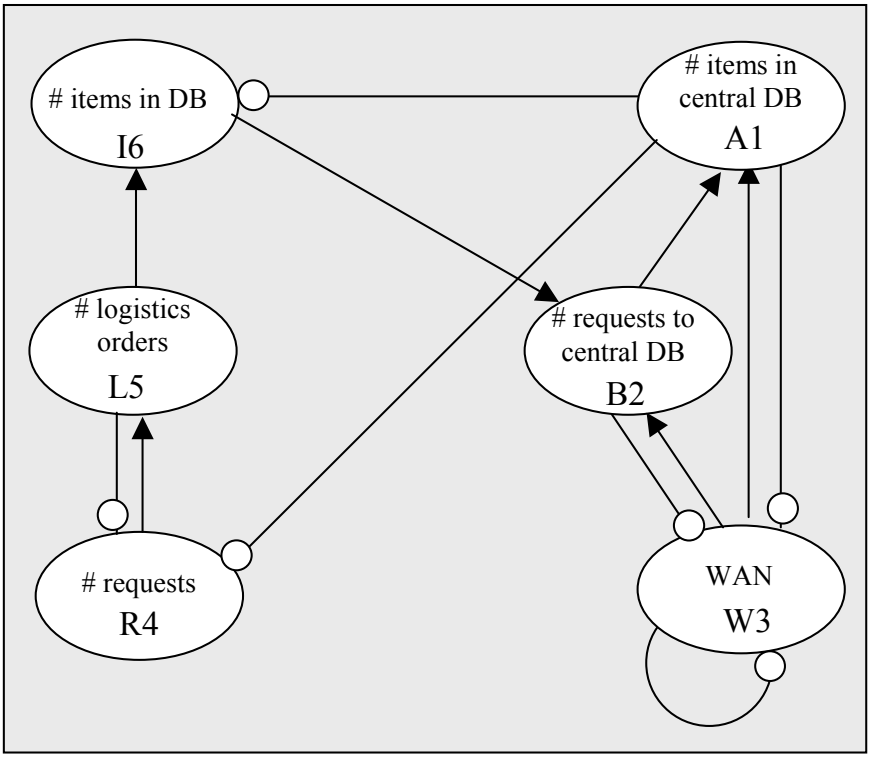


Figure 34. Signed digraph for the baseline logistics system

The results of the simulations are:

Total number of simulations:	5000
Likely stable:	78
Manageable:	2381
Likely unstable:	2541

These show that the system has a reasonable range of stability among the 5000 quantitatively specified models simulated.

The adjoint matrix is:

$$adjoint\_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & -2 & 1 & 0 & 1 \\ 2 & 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & -2 & 1 & 0 & 1 \end{bmatrix}$$

The weighted prediction matrix is:

$$weighted\_predictions\_W = \begin{bmatrix} 1. & 1. & 1. & 0 & 1. & 0 \\ 1. & 1. & 1. & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & 0 & 1. \\ 1. & 1. & 1. & 0 & 1. & 0 \\ 1. & 0 & 1. & 1. & 1. & 1. \end{bmatrix}$$

The weighted predictions matrix contains no values less than one. This means that all nodes are equally vulnerable.

In summary, the baseline system does not show a strong tendency to either stability or instability. It will be moderately manageable following an attack. All links are equally vulnerable. The system's structure does not offer protection to certain nodes nor does in confer significant recovery capabilities.

Logistic system modified to include increased low-level feedback

The baseline logistics system was modified to contain increased low-level feedback (Figure 35).

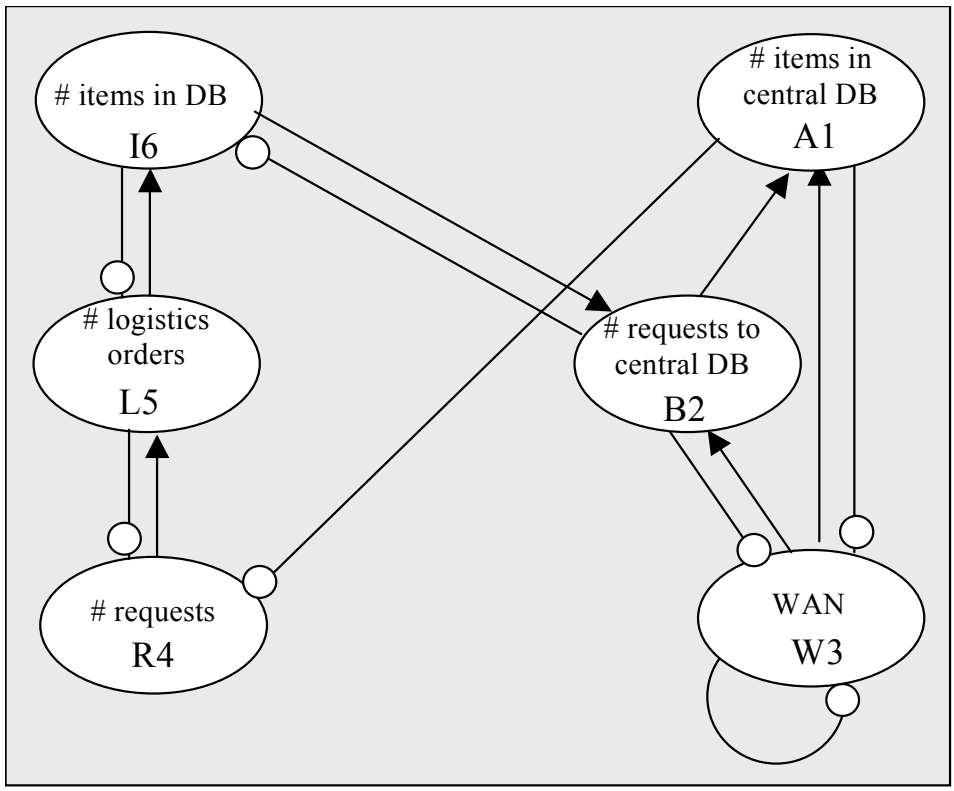


Figure 35. Signed digraph for the baseline logistics system modified with low-level feedback

This means that more decision points have been inserted into the system. There are now more points in the system under local control, where control of the process is relinquished and transferred to the next variable. One introduced point of local control in the modified system, for example, occurs at Node 5, number of items in the local database. This means that logistics orders are acknowledged when they are received into the local database and that the control of the order is relinquished to the database. The requisitioner no longer has control over the process. Rather than one long process with one entry point and one end point, the process has been divided into several sub-processes. The modified system is shown in Figure 35.

The community matrix for this system is:

$$A := \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The results of the simulations are:

Total number of simulations:	5000
Likely stable:	1370
Manageable:	2527
Likely unstable:	1103

These results show that this system has a much broader range of stable behavior in simulations than the baseline model.

The adjoint matrix is:

$$adjoint\_A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 0 & -1 & -1 & 1 & -1 & 1 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 1 \end{bmatrix}$$

The weighted prediction matrix is:

$$weighted\_predictions\_W = \begin{bmatrix} 1. & 1. & 1. & 0 & 1. & 0 \\ 1. & 1. & 1. & 1. & 1. & 1. \\ 0 & 1. & 1. & 1. & 1. & 1. \\ 0 & .33 & 1. & 1. & .33 & 1. \\ 1. & 1. & 1. & 1. & 1. & 0 \\ 0 & .33 & 1. & 1. & 1. & 1. \end{bmatrix}$$

The richer structure of this system makes some links less vulnerable, in the sense that they contribute less to the overall structure of the system. Attacks on the links with weighted prediction values of .33 will be less likely to transmit effects through the system. However, although the system is more resilient than the baseline with respect to vulnerability, it will display highly oscillatory behavior as it recovers from an attack. This can be surmised by inspection of the eigenvalues (-0.129 + 1.74i, 0.034 + 1.26i, -0.400 + .2i) that all contain complex parts.

In summary, the baseline system modified to contain increased low-level feedback shows an increased tendency to stable behavior. However, the system will oscillate. This means that delays in processing are an inherent attribute of the recovery process for this system. Weighted feedback values of .33 show areas where system structure may mitigate and deflect the effects of an attack. Structural patterns in the system make some variables less vulnerable than others.

Logistic system modified to include increased high-level feedback

The baseline system was then modified to incorporate greater high-level feedback. The modified system is shown in Figure 36.

In this system, high-level feedback reduces the number of decision points in the system. An order may traverse long paths unacknowledged and uncorrected. The intermittent variables along a path do not exert control over the process. This means that when the process is interrupted there are few opportunities for the system to adapt. Orders will continue to be submitted, unacknowledged, even though the system is broken. There is little low-level feedback (few short loops) to enable autonomous corrections to the number of logistics orders the system can accommodate.

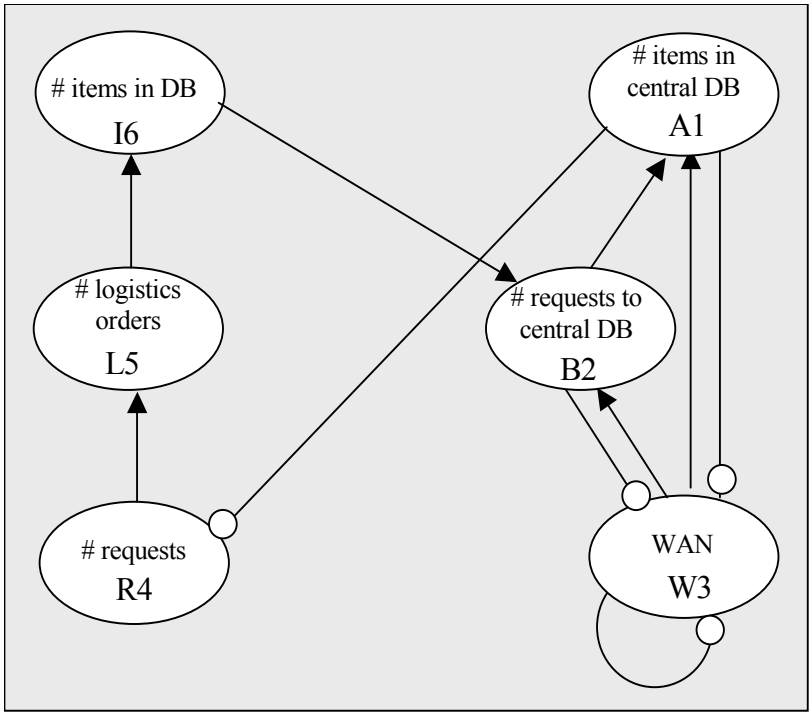


Figure 36. Signed digraph for the baseline logistics system modified with high-level feedback

The community matrix for this system is:

$$A := \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The results of the simulations are:

Total number of simulations:	5000
Likely stable:	0
Manageable:	1273
Likely unstable:	3727

These results show that this system is highly unstable. We have little reason to believe that its structure can support return to equilibrium following an attack. It will require extensive input of resources to restore links.

For those cases when stability can be achieved following perturbation, the adjoint matrix is:

$$adjoint\_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

The weighted prediction matrix is:

$$weighted\_predictions\_W = \begin{bmatrix} 1. & 1. & 1. & 0 & 1. & 1. \\ 1. & 1. & 1. & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & 0 & 1. \\ 1. & 1. & 1. & 1. & 1. & 0 \\ 1. & 0 & 1. & 1. & 1. & 1. \end{bmatrix}$$

As in the baseline model, all links are equally vulnerable and contribute substantially to the overall functioning of the network.

In summary, the baseline logistics system modified to contain increased high-level feedback has a significant tendency to be unstable. Following an attack, the system may recover, but it will require intensive management and the infusion of extensive resources during the recovery period. When the predictions apply, all variables are equally vulnerable. The structure of the system does not offer protection and contributes negligible internal structural support for recovery.

Comparative analysis of predictions

The three alternative models of the logistics system scenario shown in Figures 34, 35, and 36, respectively, represent three possible scenarios and have very different properties. The prediction matrices for the three models are shown in Figure 37. These three prediction matrices form a set of testable alternative hypotheses. A hypothesis may be refuted when the predictions for the associated model fail to match observations when the system is stressed. The stress may take the form of an attack or a monitored challenge to the system specifically designed as an experiment to test the hypothetical model.

The prediction matrices present interesting consequences. They share one property. In all cases, elements a_{23}, a_{24}, a_{33}, a_{34} have a response with high weight, indicating that these changes will occur in all cases if a press to the system were to occur on nodes 3 and 4. (The direction of the response is indicated by the sign of the matrix element.) This means that even if the exact structure of the system is not known, these changes are predictable, which can be either a strength or a weakness. These invariant predictions among the models are highlighted in pink rectangles in Figure 37.

Stability, the ability to recover from a temporary disturbance, is ranked from most to least stable, from the modified model containing increased low-level feedback (Figure 35), the baseline model (Figure 34) to the model containing increased high-level feedback (Figure 36). This behavior is positively related to the presence double links, which intuitively provide low-level or direct control. The model containing increased

high-level feedback, however, demonstrates significant tendencies for unstable behavior and will manifest as a stable system only under highly constrained conditions.

Paradoxically, the less stable models present less change following a press. Thus, three columns in the prediction matrix for the model with increased high-level feedback and the baseline model contain columns of zeroes, mostly with high weights. These columns are highlighted by blue ovals in Figure 37. These results indicate that most variables would not change in abundance, albeit those that do would do so in an unpredictable fashion, given their low weight. These variables are protected from input by the structure of the system. The most stable model, the model with increased low-level feedback (Figure 35) possesses a high number of non-zero responses. Also counter-intuitively, the least stable models have the highest weights overall.

The desirability of any model is dependent on whether or not monitoring or protection is a priority. If monitoring is essential, then the model in Figure 34 would be the structure most likely to indicate a response and would be desirable. If protection of the most variables is desirable, then the least stable system would be more desirable.

The results suggest that it is possible to construct a system with particular defensive goals in mind. If, on the one hand, the site of an attack is predictable, monitoring is not feasible and there is a requirement for protecting a large number of variables, the model containing increased high-level feedback (Figure 36) might be best. A drawback is that the strength of interactions would have to be kept within strict limits, given the potential for instability of this system. On the other hand, if attack sites are unknown, fluctuations are inevitable and recovery is a priority, then the model containing increased low-level feedback (Figure 35) would be best.

Prediction matrix for baseline model

$$adjoint\_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & -2 & 1 & 0 & 1 \\ 2 & 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & -2 & 1 & 0 & 1 \end{bmatrix}$$

Prediction matrix for model with increased low-level feedback

$$adjoint\_A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 0 & -1 & -1 & 1 & -1 & 1 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 0 & 1 \end{bmatrix}$$

Prediction matrix for model with increased high-level feedback

$$adjoint\_A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Pink rectangles indicate common response across all models.
Blue ovals show variables that do not respond to change.

Figure 37. Prediction matrices for (a) Baseline model, (b) Model with increased low-level feedback, and (c) Model with increased high-level feedback

Overall, it is possible either to construct, in advance, a system with a specific set of links that give it certain properties or, more importantly, to modify a system's links in response to a changing landscape of attacks. Model analysis also allows us to list both strengths and weakness present in any given system.

5.6.2 Propheteer Strawman scenario

We applied the system dynamics approach to variables derived from the Propheteer Strawman model to produce a plausible global system representation.

Communication infrastructure is organized into subsystems for delivery of specific modes of communication: computer networks, cellular telephone, telephone, satellites. These subsystems form a 'guild' of communication providers, a subset of variables that form one functional level of hierarchy within the global system. These providers may compete with each other and this competition may represent the adaptive nature of response to attack. For example, when conventional telephone networks are compromised, users switch to cellular telephones. This switching behavior allows the overall structure of the global system to be maintained by substituting a variable within the same guild.

Our goal was to present the mission as the product of an ecosystem consisting of resources, infrastructure and actors, delivering the output of one level of organization to higher levels in the system hierarchy. We found that the Propheteer Strawman Scenario, attack trees in general and other models are oriented towards an atomistic, event-driven description of effects and countermeasures. Our goal is to express the broader effects of input in the successful completion of the mission.

We model missions as either stabilizing or destabilizing. Stabilizing missions provide goods and services that are consistent with the normal functioning of the mission ecosystem. The ecosystem may be required to function at a higher rate during times of stress but remain stable. For example, in response to an attack or threat, stabilizing missions preserve the basic patterns of interaction within the system. Destabilizing missions are intended to disrupt patterns of interactions within the system. Usually undertaken by an enemy, a destabilizing mission might include disruption of a crucial link between two variables that impairs the ability of the system to maintain its normal functions. Destabilizing missions may be offset by adequate defense to prevent the disruption from occurring at all. This may be enhanced by prior identification of vulnerable nodes within the system. Once an attack has occurred, the effects may manifest themselves throughout the system as they cascade through feedback pathways. The locations of these indirect pathways may also be illuminated by analysis of system dynamics.

At a global level, the system producing the mission can be modeled as a competitive relationship between two actors, such as the United States and China in this scenario. At an atomistic level, the base resource of computer networks can be modeled

as a community formed by bandwidth (resource), sysadmins (infrastructure) and the functioning network (actor) that delivers a service to the larger system. These intermediate levels are much more difficult to discern and many alternate models are possible. Models vary in response to their purpose. The models contained in this report describe an ecosystem supplying goods and services. Military effectiveness is modeled as the 'high level' consumer. Military effectiveness itself is a subsystem consisting of resources, infrastructure and actors. Many of the CC2 scenarios described activities designed to affect military personnel, so we create a distinct node for this variable.

The CC2 events in Day 1 of the Propheteer Strawman scenario describe activities directed at critical national infrastructure through communications systems. They describe email threats targeted at military personnel, scanning, and insufficient available system administrator time to address all threats in a timely manner.

On Day 2, the State Department is added to the model, when CNN, a media provider, reports based on information gathered from a classified system that the United States will likely revoke China's MFN status.

On Day 5, the New York Stock exchange and the NASDAQ shut down in mid-day as a result of false stock reporting on portals such as Yahoo and AOL, causing investors to panic.

Base level model

The base level of the model consists of resources that support the other, higher-level variables. In the model presented, we take communications to be the base-level resource (Figure 38). Communications, supported by a variety of providers (telephones, cellular telephones, computer networks), enable the global system to deliver defensive, financial, and diplomatic services required by the mission.

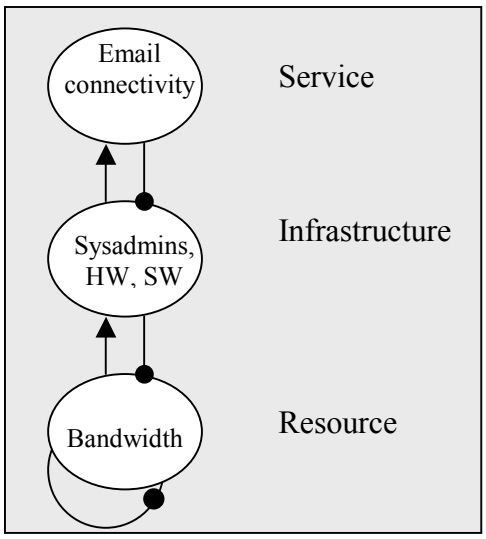


Figure 38. Communications subsystem

The links are resource/consumption links because the product of each level is consumed by the previous level to produce a good or service for the subsequent level.

Bandwidth is consumed by infrastructure to produce email connectivity. The resource, bandwidth, is self-regulated because it is produced by factors outside of this system. Infrastructure is also self-regulated, but email connectivity, which we take to be completely dependent upon resources and infrastructure, is not.

There are several alternative providers of communications, each of which contains its own resources and infrastructure. Each communication variable is summarized as a self-regulated variable. Together they form a guild of communication providers (Figure 39 – Note that for compact representation competitive links are represented by single lines terminating in bubbles at both ends rather than two separate links). All members of the guild need not be present in any one instantiation of the system. However, among those that are present, competitive links allow one variable to assume a dominant relationship in the system when another competing communications variable is depressed. This allows for users to switch from conventional telephones to cellular telephones, for example, when conventional telephone networks are inconvenient or nonfunctional.

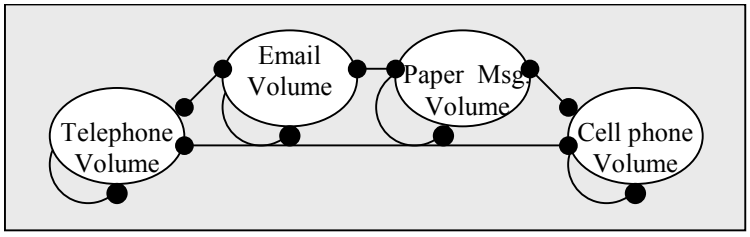


Figure 39. Communications guild

The entire guild, given it has overall negative feedback, can then be collapsed into one system (Figure 40):

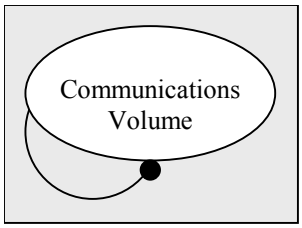


Figure 40. Communications variable

The system can be expanded for further analysis after it has been identified as a vulnerable node in the larger system.
Media are also variables in the global model. Specifically, CNN and portals are called out in the Propheteer Strawman scenario. We model media leverage, the ability of the media to influence based on credible reports, as a subsystem of the global network (Figure 41).

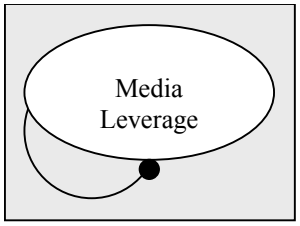


Figure 41. Media variable

False reports from portals caused the New York Stock Exchange NASDAQ to shut down on Day 5 of the scenario. We model the NYSE volume as a self-regulated variable, since it is supported by brokers and analysts outside of the system. Investors consume the product of the NYSE, but are also modeled with self-regulation, meaning that they are not completely dependent upon it. We also depict the relationship between media and NYSE as resource/consumer, since the NYSE is able to increase its volume due to wide-ranging participation enabled by media reporting of stock prices (Figure 42).
Magnitude of the insider threat (Insiders) is represented in the model as a variable. By incorporating them as a variable, their presence is modeled as a PRESS experiment, as a permanent change to the system.
China is included in the diagram, but is unlinked to the system under study. This discontinuity is indicated by a "?" in the digraphs. Although it may be the origin of many of the inputs to the system, the variable itself exists at a higher level of aggregation and, for the purposes of this modeling exercise, remains relatively undefined and nebulous.

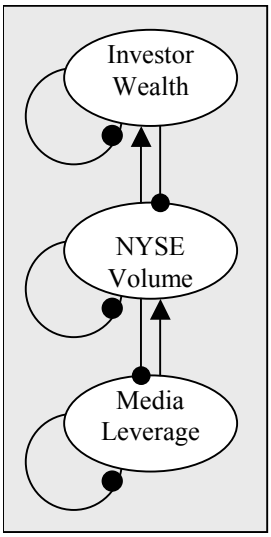


Figure 42. Financial subsystem

On Day 2, an insider in the State Department leaked a classified message to CNN.
We define the variable State Department Leverage (State Dept.) as the ability of the State Department to effectively influence matters of state. We link the variable NYSE to State Department in a comensal relationship, since overseas investments are tied to positive US relationships in foreign countries.
Since the email threats were directed at military personnel, we include personnel as a variable in the system (Figure 43).

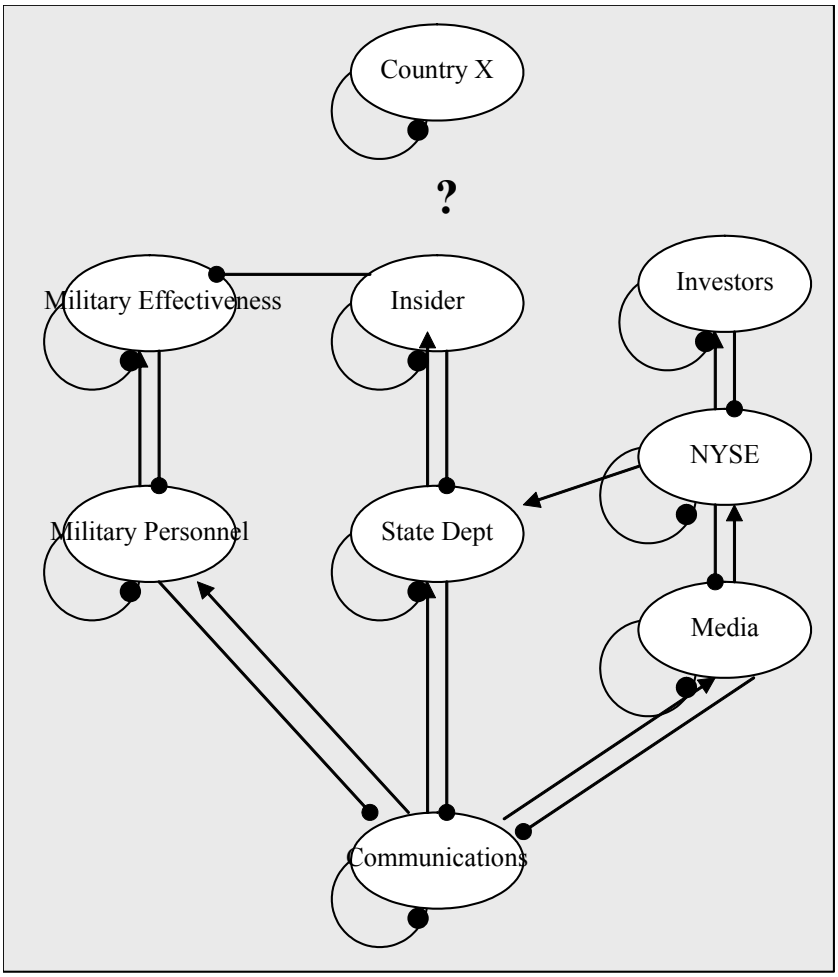


Figure 43. Propheteer Scenario digraph 1

Variables

- Communications = Communications Volume
- Military Personnel = Availability of military personnel
- Military Effectiveness
- State Dept = State Department leverage
- Insider = Magnitude of Insider Threat
- Media = Media Leverage
- NYSE = NYSE Volume
- Investors = Investor Wealth

With the variables selected, we are now able to assemble them into a community by linking them together using arrows and bubbles to describe positive and negative relationships, respectively. Recall that an arrow from one variable to another indicates that an increase in the first variable causes an increase in the second. Consumption of a resource provided by a variable is depicted by an arrow out of that variable and a bubble into the same variable.

Analysis

The global system shown in Figure 43 can now be used to generate hypotheses regarding system vulnerabilities and response. Attacks can be identified by their point of entry into the system and their indirect effects can be traced through the system.

The community structure is summarized in a matrix consisting of i=j rows and columns, effectively. A value of positive 1 indicates a positive relationship to the ith variable from the jth variable. A value of –1 indicates a negative relationship. The matrix representation of this community is:

The variables occur in the order:

1. Communications = Communications Volume
2. Military Personnel = Availability of military personnel
3. Military Effectiveness
4. State Dept = State Department leverage
5. Insider = Magnitude of Insider Threat
6. Media = Media Leverage
7. NYSE = NYSE Volume
8. Investors = Investor Wealth

Community matrix for Figure 43:

$$A := \begin{bmatrix} -1 & -1 & 0 & -1 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

The adjoint is shown below. Recall that the adjoint is used to derive the predicted behavior of the system to a PRESS, or sustained input. The magnitude of the entries in the adjoint is used in the weight calculations. The predicted effect of an input into the jth column variable on the ith row variable is given by the sign of the a_{ij}th matrix element.

Adjoint matrix for Figure 43:

$$adjoint\_A = \begin{bmatrix} 12 & -6 & 6 & -9 & 3 & -11 & 1 & -1 \\ 10 & 14 & -14 & 2 & 12 & -6 & 4 & -4 \\ 2 & 18 & 20 & -11 & -9 & -5 & -3 & 3 \\ 8 & -4 & 4 & 13 & -17 & -1 & 7 & -7 \\ 8 & -4 & 4 & 13 & 21 & -1 & 7 & -7 \\ 8 & -4 & 4 & -6 & 2 & 18 & -12 & 12 \\ 4 & -2 & 2 & -3 & 1 & 9 & 13 & -13 \\ 4 & -2 & 2 & -3 & 1 & 9 & 13 & 25 \end{bmatrix}$$

The columns represent the qualitative direction of change in all variables following a press to a variable (corresponding to the column number).
This particular system can be very stable because of its strong self-regulation and many resource/consumption links. In the system specified, communications and military personnel are the two most vulnerable nodes. Weights of .5 or above are considered very reliable, from the weighted predictions matrix below. Each element gives the weight of the corresponding element in the adjoint matrix above.
Qualitative analysis of the structure of this system yields the following weighted predictions matrix (here with the sign of direction following a positive input):

Weighted predictions matrix for Figure 43:

$$weighted\_predictions\_W = \begin{bmatrix} 1. & 1. & 1. & 1. & .33 & 1. & .14 & .14 \\ 1. & 1. & 1. & .25 & 1. & .75 & .67 & .67 \\ .20 & 1. & 1. & 1. & .60 & .56 & .43 & .43 \\ 1. & 1. & 1. & 1. & 1. & .14 & 1. & 1. \\ 1. & 1. & 1. & 1. & 1. & .14 & 1. & 1. \\ 1. & 1. & 1. & 1. & .33 & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & .33 & 1. & 1. & 1. \\ 1. & 1. & 1. & 1. & .33 & 1. & 1. & 1. \end{bmatrix}$$

The weighted predictions show the contribution of structure to the qualitative predictions. A prediction w_{ij} with a value of 1 in the weighted prediction matrix is unambiguous and

reliable with respect to system structure. When the value of a weighted prediction falls below 0.5, the probable sign of the prediction is structurally ambiguous and unreliable (Dambacher 2002). This does not mean that the quantitative instantiation of the system is guaranteed to behave in the manner specified by the prediction matrix. Rather, the weighted predictions show which behaviors are enhanced by structure. More importantly, the qualitative and weighted prediction matrices show behaviors that might be precluded by system structure.

Table 11. Table of predictions with weights for Propheteer Strawman digraph 1

Input to:

Effect on:	Communi-cations	Availability of military personnel	Military effective-ness	State Department leverage	Magnitude of insider threat	Media leverage	NYSE volume	Investor wealth
Communi-cations	+ (1)	- (1)	+ (1)	- (1)	+ (.33)	- (1)	+ (.14)	- (.14)
Availability of military personnel	+ (1)	+ (1)	- (1)	+ (.25)	+ (1)	- (.75)	+ (.67)	- (.67)
Military effective-ness	+ (.20)	+ (1)	+ (1)	- (1)	- (.60)	- (.56)	- (.43)	+ (.43)
State Department leverage	+ (1)	- (1)	+ (1)	+ (1)	- (1)	- (.14)	+ (1)	- (1)
Magnitude of insider threat	+ (1)	- (1)	+ (1)	+ (1)	+ (1)	- (.14)	+ (.1)	- (1)
Media leverage	+ (1)	- (1)	+ (1)	- (1)	+ (.33)	+ (1)	- (1)	+ (1)
NYSE volume	+ (1)	- (1)	+ (1)	- (1)	+ (.33)	+ (1)	+ (1)	- (1)
Investor wealth	+ (1)	- (1)	+ (1)	- (1)	+ (33)	+ (1)	+ (1)	+ (1)

The direction of the effect due to positive input into any variable is shown as 1, 0, or –1 in Table 3. The weighted predictions that describe the contribution of system structure to the effects of system input are shown in parentheses next to the predicted direction of effect. Vulnerabilities may be read by row or column as column entries with a negative sign and weight above 0.5. This threshold has been validated in simulation studies (Dambacher 2000). When reading across a row, the vulnerability is read as the variables that will affect the row variable. When reading down a column, the vulnerability is read as the effect of input into the column variable on the row variable.

From Table 11, reading down the columns, input into variables 2 (availability of military personnel) and 4 (State Department leverage) will negatively impact the maximum number of variables. Positive input into the military personnel (i.e., more time available) leads to decreases in:

- Communications – as more resources are consumed to do work;
- State Department leverage – as resources are devoted to military matters, rather than those of State;
- Insider threat – as resources are diverted to detection and eradication or insider threat;
- Collateral decreases in media leverage, NYSE volume, and investor wealth – as effects filter through the system.

The attractiveness of the news media as a terrorist target was noted on 14 October 2001, by United States Attorney General John Ashcroft (2001): "If I were a terrorist, I would want to engender fear that was irrational, and I would want to curtail the availability of information in a free press that was good information."

Reading across the row, military effectiveness, we can read off the variables that will negatively impact military effectiveness. Increased State Department leverage and increased magnitude of insider threat will negatively affect effectiveness, as will media leverage.

Addition of a mutualistic link between State Department leverage and military effectiveness changes the vulnerabilities present in the system (Figure 44). The coefficients of the characteristic polynomial are negative, which signifies that the system has sufficient negative feedback to support stability. The predictions are shown in Table 12. The addition of this mutualistic, mutually enhancing relationship has removed many vulnerability points. Increased magnitude of insider threat will cause decreased military effectiveness and State Department leverage, however, many of the other strong weights have been dissipated through this system modification. The modification does not come without a price, however. Along with decreased vulnerability (defined as targets that will most likely cause the most structural damage to the system when affested), comes decreased predictability. Increased complexity has made the system 'softer'; its response to input will now be more ambiguous (less predictable).

The variables occur in the order:

1. Communications = Communications Volume
2. Military Personnel = Availability of military personnel
3. Military Effectiveness
4. State Dept = State Department leverage
5. Insider = Magnitude of Insider Threat

6. Media = Media Leverage
7. NYSE = NYSE Volume
8. Investors = Investor Wealth

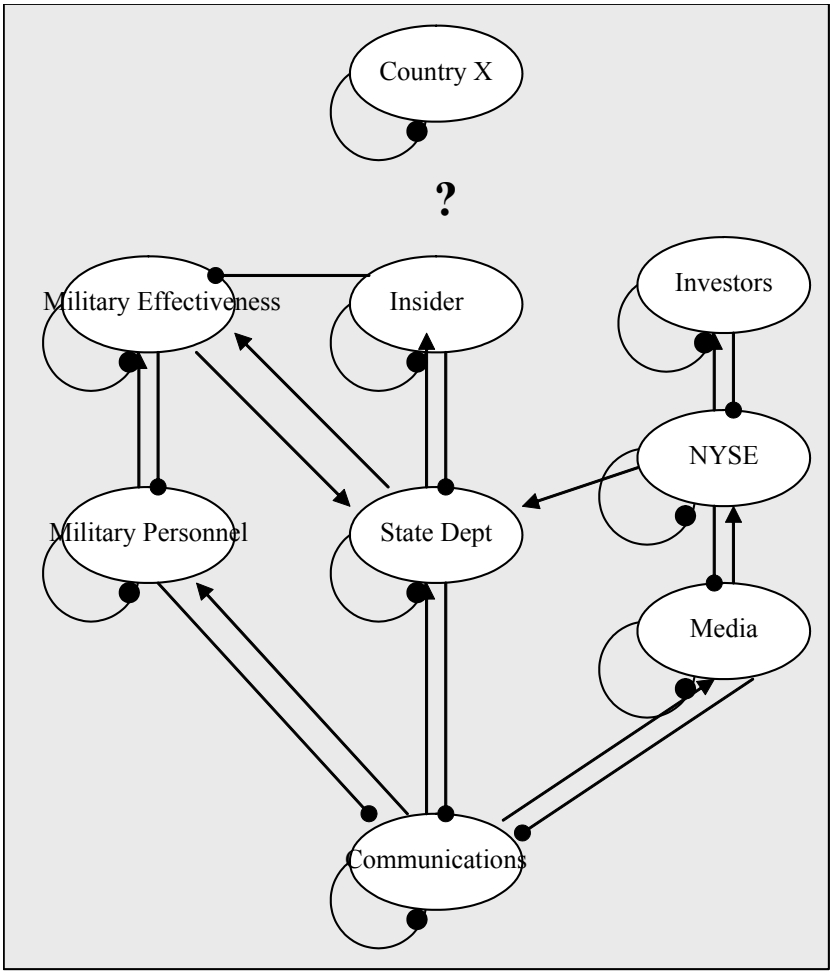


Figure 44: Propheteer Strawman digraph 2

Table 12. Table of predictions with weights for Propheteer Strawman digraph 2

Input to:

Effect on:	Communi-cations	Availability of military personnel	Military effective-ness	State Department leverage	Magnitude of insider threat	Media leverage	NYSE volume	Investor wealth
Communi-cations	+ (.67)	- (.60)	+ (.33)	- (.50)	+ (.20)	- (.63)	+ (.20)	- (.20)
Availability of military personnel	+ (.30)	+ (.58)	- (1)	- (.23)	+ (1)	- (.33)	+ (.091)	- (.091)
Military effective-ness	+ (.43)	+ (.64)	+ (1)	- (.16)	- (.74)	- (.38)	+ (.0.91)	- (.091)
State Department leverage	+ (1)	+ (.11)	+ (1)	+ (1)	- (1)	- (.33)	+ (1)	- (1)
Magnitude of insider threat	+ (1)	+ (.11)	+ (1)	+ (1)	+ (.33)	- (.33)	+ (.1)	- (1)
Media leverage	+ (.67)	- (.60)	+ (.33)	- (.50)	+ (.20)	+ (.60)	- (.58)	+ (.58)
NYSE volume	+ (.67)	- (.60)	+ (.33)	- (.50)	+ (.20)	+ (.60)	+ (.62)	- (.62)
Investor wealth	+ (.67)	- (.60)	+ (.33)	- (.50)	+ (.20)	+ (.60)	+ (.62)	+ (.60)

5.6.3 Ecological model of a DDoS attack (Code Red)

In previous work we modeled malicious code as a disease that was distinct from the community it disrupts. In this epidemiological approach, we evaluated the impact of the malicious code in terms of basic reproduction rate and generation time. It is useful when the information obtained from the analysis can be used to slow or eliminate the infection. When the attack proceeds so quickly that no such reaction is possible, we must look ahead to the effects of the attack. In this ecological approach, we model the community and its predicted response to attack as it is mediated by all of the interacting, preexisting variables present in the system.

Attacks may be modeled as pulses or presses. As we discussed previously, pulses deliver a measured dose of input and then cease. In a pulse experiment, a reasonable input is applied, and the community is observed as it returns to its perturbation equilibrium state. In a press experiment, the input is incorporated permanently (as a modified input or output rate) into the community. Vector-borne disease can be viewed as a press experiment with at least three variables, disease organism, vector, and host. More complicated transmission communities are possible and these have been discussed in Chapter 4.

In Figure 45, we model an ecological system in which Code Red is a 'top predator'. The variables in this system are:

- available bandwidth
- number of machines running IIS not infected by Code Red
- number of machines running Apache (not infectable by Code Red)
- level of usage (work capacity of the system)
- number of machines running IIS infected with Code Red.

Inclusion of Code Red as an element in a stable community is supported by recent reports that Code Red II continues to infect, months after its initial release. Dug Song, a security architect at Arbor Networks noted, "Code Red and Nimda are going to be a permanent part of the Internet landscape for some time to come" (Costello 2002).

All variables, except Code Red, are self-regulated because they receive contributions from outside this specific system. One way to test the self-regulation assumption is to ask: If the variables providing input into this variable were eliminated, would the variable still exist? If the answer to this question is affirmative, then a self-regulation link on the variable under consideration should be included in the model. Code Red, which exploits an IIS vulnerability, is modeled as a negative input into IIS. Since Code Red is not maintained by resources outside the system and since it does not exhibit density-dependent growth (in fact, its purpose is to replicate to the maximum extent possible, without bound), it does not receive a self-regulation link. The signed digraph for this model is shown in Figure 45.

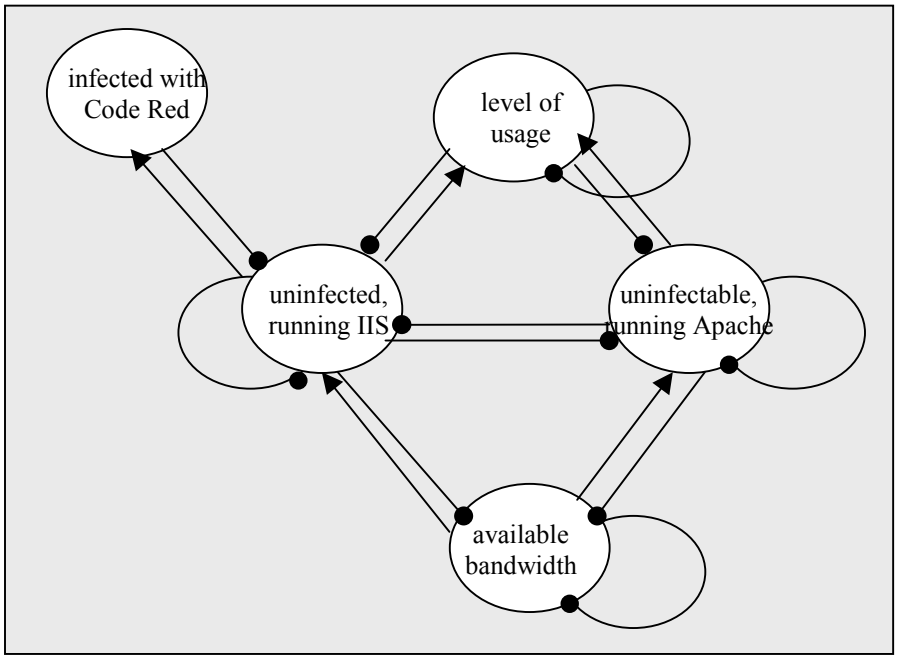


Figure 45. *Code Red* community level model

As in any modeling exercise, clear and concise definitions of variables and their interrelationships are necessary to construct an accurate model. The process of generating a model like the one shown in Figure 45 is summarized in Table 13.
The signed digraph of the system (Figure 45) is then converted into a matrix representation. Each column of the resulting inverse matrix represents the change that will occur for each variable when a positive input occurs in the variable corresponding to this column. The effect of input on all variables from all possible input is thus tabulated producing a *table of predictions* (Table 13).
Column two shows the effect of control through depression of IIS. The table entry in column two, row five (a_{52}) shows the effect of a positive input into the number of uninfected machines running IIS on the number of machines infected with Code Red. To determine the effect of a negative input, the sign in this table entry is reversed, so that we read that a negative input into the number of uninfected machines running IIS causes a decrease in the number of machines infected with Code Red. Control is achieved with little disruption to the existing system structure by reducing the number of servers running IIS. This is not necessarily a viable alternative, but represents one result of the analysis. An alternative means of control through increased utilization of Apache servers is shown in column 3. Increased numbers of Apache servers result in a reduction of the number of machines infected with Code Red (column three, row five) utilized bandwidth (column three, row one) with no change in number of IIS servers (column three, row two). Input into Code Red alone is not sufficient to curb its effects. Reduction in the number of machines infected with Code Red (column five, reverse signs for negative input) increases the number of uninfected machines running IIS, but does not decrease the number of machines infected with Code Red (column five, row five). These results

reflect the interconnectedness of the system. When Code Red is taken as a community variable delivering a protracted effect on the system, its effects are mediated by all community members. It is no longer possible to isolate and contain the threat.

Table 13. *Code Red* community level model prediction matrix

INPUT→ **OUTPUT↓**	Available bandwidth	Number uninfected, running IIS	Number uninfectable, running Apache	Level of usage	Number infected with Code Red
Available bandwidth	+ (increase)	0 (no change)	- (decrease)	+ (increase)	0 (no change)
Number uninfected, running IIS	0 (no change)	0 (no change)	0 (no change)	0 (no change)	- (decrease)
Number uninfectable, running Apache	+ (increase)	0 (no change)	+ (increase)	- (decrease)	+ (increase)
Level of usage	+ (increase)	0 (no change)	+ (increase)	+ (increase)	0 (no change)
Number infected with Code Red	0 (no change)	+ (increase)	- (decrease)	0 (no change)	0 (no change)

The ecological approach to the analysis of cyber attack is a novel one. While many current control technologies focus on the most basic components of the system such as firewalls and routers, the ecological approach acknowledges that some attacks will overwhelm these control measures and allows administrators and analysts a way to look ahead to the consequences of such a breach. The approach forms a complement to current detection technologies.

5.6.4 A CPU-centric model for availability (Morris worm DDoS attack)

An attack on the availability of a system to perform its work (essentially a denial-of-service) occurs when the resources of the system are insufficiently available to perform the normal work of the system. We model this by considering the available resources of the system as a variable. In our model, the resource is available CPU time, and we posit that normal work consumes available CPU time in a predator-prey fashion.
This model represents worm propagation along the lines of the Morris worm (Reynolds 1989). Out of confidentiality, integrity, and availability, the Morris worm essentially attacked availability (because it used so much CPU time on machines that it infected). Thus, this model is centered around available CPU time; however, one could incorporate other resources into this variable as well. This model applies to other resource-hungry worms such as Code Red.

Variables

We define a CPU-centric model, in which we have the following quantities. We note that this model can be interpreted at the level of one computer, or an entire network.

- **Available CPU time:** The available CPU time is a regenerating resource, which is consumed by any kind of work performed by the computer.
- **Normal work:** Normal work is work done by the computer system in the process of fulfilling its mission, such as user applications.
- **Abnormal work:** Abnormal work is work done by unauthorized or otherwise anomalous processes, such as worms that are trying to crack passwords or to spread to other machines.
- **Vulnerabilities:** Vulnerabilities of the system(s) can be exploited (by crackers, unauthorized users, worms, viruses, etc.) to create abnormal work on the system.
- **Countermeasure effort:** Countermeasure effort is administrator effort used to patch vulnerabilities.

Relationships

We document and justify relationships between variables.

- Available CPU time is self-regenerating and thus has a negative self-effect. Also, available CPU time is consumed by both normal and abnormal work. Furthermore, available CPU time is consumed by countermeasures. This relationship depends on the assumption that administrators must reduce available CPU time to install patches, for instance because they have to shut down machines.
- Normal work has a negative self-effect, since its persistence is controlled by outside influences, such as mission requirements and user needs. Normal work is a consumer of CPU cycles.
- Abnormal work is a consumer of CPU cycles. Abnormal work is increased when vulnerabilities increase, since vulnerabilities increase opportunities for malicious elements to insert abnormal work into the system. Abnormal work on the system causes the administrators to increase effort reacting to abnormal situations, thus raising the amount of countermeasure effort.
- Vulnerabilities are self-regenerating, since they come into existence by means of parameters outside the system (when they are uncovered or inserted). Increasing countermeasure effort causes vulnerabilities to decrease, since the administrators are expending effort on patching vulnerabilities.
- Admin countermeasure effort preys on CPU time, assuming the admin has to reduce the availability of the CPU to install patches (e.g., by shutting down the machine). Admin countermeasure effort increases with abnormal work, due to poorer machine functionality, user complaints, alarms, etc., which presumably causes the admin to devote time fixing vulnerabilities.

  The digraph for this model is shown in Figure 46.

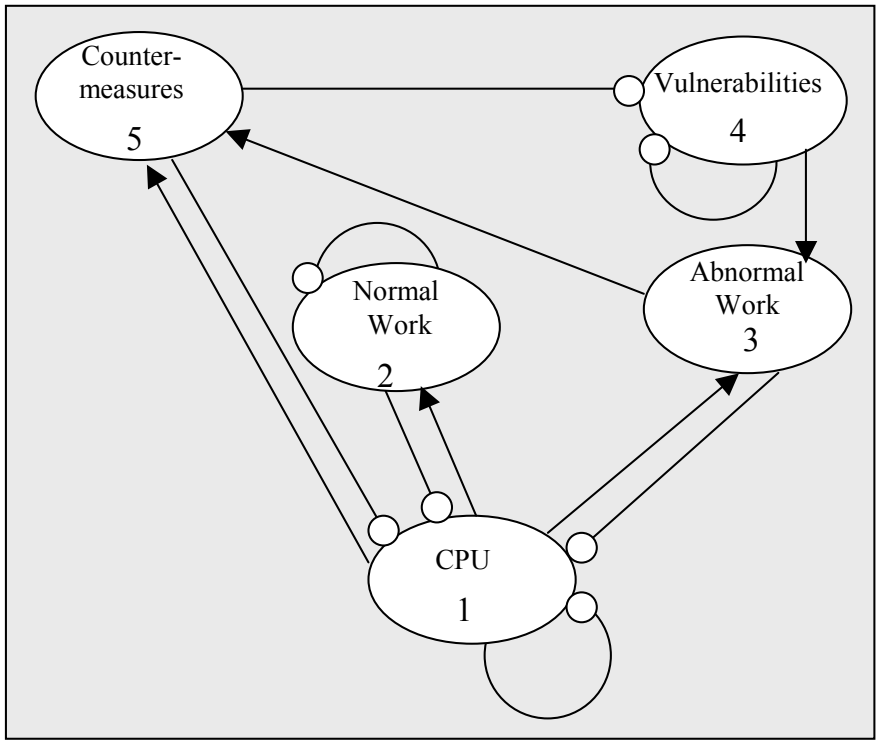


Figure 46: CPU-centric model of availability attack

Predictions

We model the Morris worm as a positive press perturbation on abnormal work, consistent with how the Morris worm was first perceived. Another way to interpret this press is as an increased awareness among malicious elements of how to exploit the existing vulnerabilities, leading to increased exploitation. This model accounts for the appearance of copycat worms that take advantage of similar vulnerabilities, which has happened frequently after the initial release of a worm (such as Code Red). The release of a worm can be perceived not only as a one-time pulse event when it is released into the wild, but also as a long-lasting effect arising from publicized vulnerabilities.

Predictions of a positive press on abnormal work are that over the long term, available CPU time will decrease, normal work will decrease, abnormal work will increase, vulnerabilities will decrease, and admin countermeasure time will increase. We believe that all of these predictions are consistent with the long-term observed behavior of systems after a publicized exploitation via a worm's release.

Here is an example of a result from the simulation tool.

```
Adjoint Matrix:
  1.00  -1.00  -1.00  -1.00   1.00
  1.00   1.00  -1.00  -1.00   1.00
 -1.00   1.00   1.00   1.00  -3.00
 -1.00   1.00  -1.00   1.00  -1.00
  1.00  -1.00   1.00   1.00   1.00
```

Total # of simulations = 5000
likely stable= 63
manageable= 261
likely unstable 4676

Note that most instantiations of this system are likely to be unstable. An inspection of the system reveals that all of the paired loops are conjoint. There are three loops of length 4, all of which pass through available CPU time, and are positive. Thus, the high-level positive feedback tends to dominate the system, resulting in instability. Our interpretation of this instability is that the system is unlikely to return to equilibrium without external intervention; for instance, a significant increase in countermeasure effort. This is consistent with the behavior of spreading worms, where sysadmins often have to make very active efforts to shut down propagation.

5.7 Examples of positive feedback systems

In this section of the report, we discuss modeling confidentiality and integrity attacks as systems dominated by positive feedback. An attack on the confidentiality of a system means that an unauthorized person is able to read or take advantage of information stored within that system. Ecologically, a model of the system would account for impact on system behavior. After our analysis of this situation, we discuss why we believe it is appropriate to model integrity attacks with the same model.

5.7.1 Confidentiality attacks

Damage from a confidentiality attack is the result of the act or threat of unauthorized disclosure. The disclosure must cause harm to the "interests" of the system in some way.
A disclosure of system vulnerabilities could allow hackers to disrupt other system work.
A disclosure of payroll information in a company may impact morale as employees bicker about salary inequities.
A disclosure of medical information could cause patient embarrassment or prevent them from getting a job or an insurance policy.
A disclosure of business trade secrets could negatively impact a company's ability to generate revenue using its proprietary methods.
A disclosure of military information could cause an operation to fail.
Various mechanisms for released confidential data could impede normal work. Thus, we choose to model confidentiality as confidential data being used to accomplish the normal work of the system. We can model attacks as disruptions of this performance of normal work.
If we model work in such a way that data must be confidential for the work to be completed properly, then we can model confidentiality violations by diminishing the pool of confidential data that can be used to accomplish normal work.
Recall the paradigm that we have been using to model the use of data as a resource; namely, modeling the "unused" data as the variable, so that performing work with it diminishes it. However, confidential data that have been used to generate normal work, but are still available in archives, could potentially have deleterious effects if released. Thus, the action of performing normal work does not really decrease the level of

confidential data. In fact, in this model, we assume that work done using confidential data actually spawns more confidential data. This is an example of positive feedback. Another aspect to model is the role of vulnerabilities in confidentiality violations. Vulnerabilities can take several forms; network vulnerabilities that allow hackers to download proprietary information are one form. Poor policies or controls that allow unauthorized access to data are another. In any case, the presence of vulnerabilities should increase the level of attack on confidentiality and increased countermeasure work done.

Now, to describe why we use the same model for integrity attacks, we first define an attack on the integrity of information in a system as an attack that corrupts the information so that it is no longer sufficiently accurate or complete for its intended use. Because our model for confidentiality attacks represents the attack on confidentiality of data as an attack on its suitability for its intended use, we found that the same model captures the essential features of attacks to data integrity. This follows because we have modeled both confidentiality and integrity of data as necessary elements to complete normal work successfully. To illustrate how these two qualities are related to each other, suppose a confidential target list is leaked, and the enemy moves its targets so that the operation does not achieve its objectives. Then the end result is the same as if an attack had corrupted the target list, thus perpetrating an integrity attack on the target list. By moving the real-world targets (through the use of the confidential data), the enemy has, in effect, attacked the integrity of the target list data, by destroying the data's suitability for its intended use.

Our graphical model is shown in Figure 47.

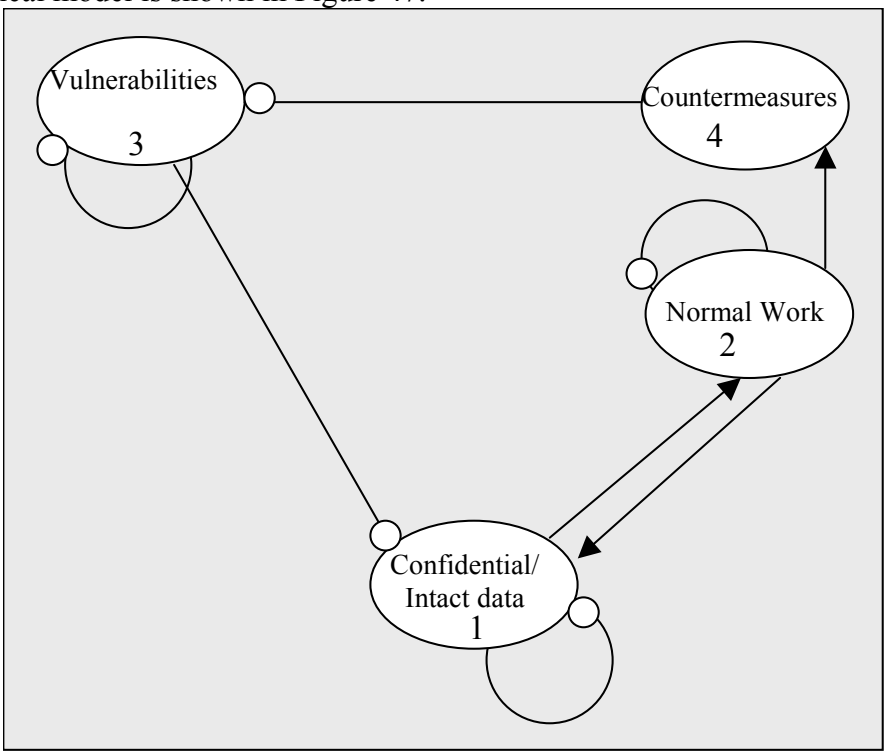


Figure 47: A general model of confidentiality attack

We assume that this system has positive feedback, and thus the conventional signs of the prediction matrix are reversed. Positive feedback is a reasonable assumption when dealing with human activities, such as economic networks or military activity. Positive feedback reflects the fact that these activities rely on resources that are increasing at a steady albeit small rate over a reasonably long period. In both economic development and warfare, lack of growth is regarded as stagnation while constant growth is perceived as stability. Positive feedback reflects this reality. Negative feedback still plays an important role. A hierarchical approach to modeling would be to identify stable sub-systems with negative feedback and assemble them when required to take advantage of a positive input over a given period.
The adjoint matrix for this system is

```
0   1   0   1
0   0   0   1
-1  -1  0   0
1   1   -1  0
```

A positive press on vulnerabilities has a positive effect on the countermeasures taken. Note however that the effect is indirect. This behavior is consistent with the following scenario: an increase in vulnerabilities leads to more attacks on confidential data, which causes the system to compensate by increasing its level of countermeasure activity. Also, a negative press on normal work (variable 2) has the following effect: a positive effect on confidential data, no effect on the level of normal work, a negative effect on vulnerabilities, and a positive effect on countermeasures. (Recall that in a positive feedback system, the signs of the predictions are opposite to those coming from a negative feedback system.) This is consistent with what happens when normal work is disrupted by loss of confidential data. After such a disruption, the sysadmins are called to take countermeasures, which reduce the level of vulnerabilities. The level of confidential or intact data recovers as it is replenished by normal work. When we ran the simulation using the community analysis module, we obtained the following result.
Total # of simulations = 5000
likely stable= 192
manageable= 159
likely unstable 4649
Note that this system also has a high-level positive feedback loop going through all of the variables, which is responsible for the system's instability.

5.8 Building structurally stable networks

Straight chain hierarchies are structurally very stable. However, their lack of redundant pathways for resource flow makes them prone to oscillation. Even though an attack may not be catastrophic, the oscillations introduced may severely impair recovery. Emergent properties that describe the health of a system, such as a tendency to oscillate, will be discussed in future work.
Given that web-like models have some advantages, how might one construct such a model with adequate foresight to enhance desirable system properties? One approach is

to specify the nature of relationships explicitly. One such disclosure can be found at http://www.wiretrip.net/rfp/policy.html.
The policy explicitly describes the relationship and responsibilities of originators of a vulnerability or problem, and the maintainer of the cognizant software, hardware or resources. The policy outlines a procedure for actions to be taken and compensation (i.e., credit for the discovery) for the discovery and resolution of a problem.
The policy does not describe a stable system, but rather specific relationships within a larger system. The interaction between originator and maintainer is a mutualistic one and potentially unstable, where the reports and credit are exchanged. Maintainers consume reports and give credit; originators consume credit and give reports. Users consume benefit from both originators and maintainers while originators and maintainers suffer loss of time. The users variable has the potential of increasing the stability of the system. This system is shown in Figure 48.

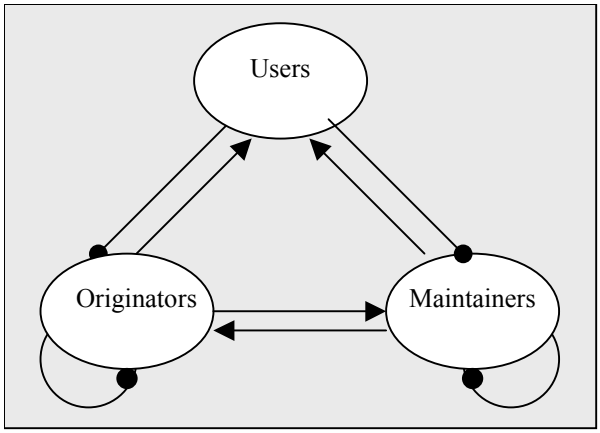


Figure 48. Keystone system defined by policy

This system is referred to as a 'keystone predator' system in ecology.

5.9 Conclusion

The systems presented in this report have been simplified for purposes of illustration. The specification of more complex systems will require domain expertise of security and network administration personnel who possess an understanding of the interrelationships among pertinent variables. The models may be focused at the security level, or may incorporate broader levels of aggregation. These broad models will benefit from the contributions of professional expertise from all areas represented.
The purpose of the models is to generate testable hypotheses. For a given system configuration, pulse and press experiments will produce measurable results that may verify system structure. These experiments manifest as attacks. Subsequent observation of the damage they cause and the success of recovery efforts will elucidate system structure, stability, and vulnerability.
In summary:

- Self-regulated straight-chains are the most stable (most likely to recover from a disturbance) system configuration, but tend to oscillate following a disturbance.
- Size of system is correlated with inherent system characteristics.

- Systems may be composed of multiple subsystems.
- Systems may be complicated horizontally or vertically.
- The adjoint of the qualitative matrix gives location of potential vulnerability, but not magnitude. 'Weight' allows assessment of potential impact.

5.10 Applications

5.10.1 Vulnerability assessment to terrorist threat

Schudel *et al.* (1999) noted the indirect nature of possible attacks: "[t]he directed adversary will attack only what is necessary to achieve their goal. If their mission can be achieved by attacking peripheral systems or even systems out of the target's control, they will."

Vulnerability is important because the "adversary will attack relatively insecure host platforms (ibid.)." Identification of host platforms that are integral to the functioning and stability of the larger system is important to prioritize the implementation of defensive measures. This is a form of preventative triage.

5.10.2 System management

One desirable management practice is to manage the community so that all components benefit. In fact, while consistent values (i.e., all positive, meaning that all variables in the community increase in tandem) in the prediction matrix are possible (Nakajima 1992), they are rare. Management regimes often must cope with prediction matrices that contain a mixture of positive and negative effects and rapidly choose which components should be enhanced.

Analysis of the underlying structure of computer networks as communities highlights network vulnerabilities through the prediction matrix. The predictions highlight variables that are vulnerable to systemic failure through indirect effects mediated through the network. Vulnerable strategic nodes may require alternate linkage patterns with the network or additional defensive measures.

5.10.3 System response

JV2020 (Shelton, 2000) states that "[b]y developing and using approaches that avoid US strengths and exploit potential vulnerabilities using significantly different methods of operation, adversaries will attempt to create conditions that effectively delay, deter, or counter the application of US military capabilities." Inducing perturbation to a vulnerable variable of a computer network community may cause many indirect effects. This press-mode of attack cannot be directly countered once the systemic changes have been initiated. Pulse-mode attacks, those resulting from the direct application of input to a variable are easier to counter. One need only remove the source of input. Press-mode attacks to an infrastructure prepared for pulse-mode response may have potentially widespread and catastrophic effects because the necessary adaptations to the continued presence of the threat are not addressed. The attack is asymmetric in this case because the attacker possesses more information than the defender regarding the intent and actual target of the attack.

5.10.4 Intelligence gathering

Just as press and pulse experiments can be applied to friendly networks to discern structure and points of vulnerability, we believe that these experiments applied to enemy networks may elucidate similar information. Adversary systems may differ fundamentally from our own. For example, Shai Feldman of the Jaffee Center for Strategic Studies at Tel Aviv University in Israel noted, "Whether it's wise or not, or appropriate or not, in reality, countries tolerate a certain level of terrorism, and live with a certain level of it" (in Purdum, 2001). Because of these differences, system response to attack may be expressed in unexpected ways. Qualitative system models constitute one method of discovering these differences and predicted response to attack. The application of these models to enemy cyber systems is beyond the scope of the current work.

5.11 Suggested research directions

The next logical step in developing the capability for rapid hypothesis generation and assessment for cyber systems using qualitative modeling techniques is to validate the models using simulated models. The models must incorporate a sufficient number of higher order variables to allow for generation of hypotheses about the entire system. The simulation-based models could be used to test the speed with which the qualitative models can be produced and their accuracy. Accuracy may be assessed by the level of agreement between the predictions and the observed simulated effects of cyber attack. The model building techniques presented in this report remain primitive. Detailed, large-scale scenarios will allow development of detailed guidelines for analysts.

5.12 References

Ashcroft J (Interview). 2001. CBS News - Face the Nation, Part I. 14 October 2001. http://www.cbsnews.com/now/story/0,1597,314724-412,00.shtml

Bender EA. 1984. Perturbation experiments in community ecology: theory and practice. *Ecology*; 65: 1-13

Carpenter T (Principal Investigator). 2001. Propheteer Strawman Sequence. Shai Technologies, Seattle, WA.

Costello S. May 8, 2002. 'Nimda,' 'Code Red' still alive and crawling. http://www.cnn.com/2002/TECH/internet/05/08/nimda.code.red.idg/index.html

Dambacher J. 2001. Qualitative analysis of the community matrix. Doctoral Dissertation. Oregon State University. Corvallis, OR.

Dambacher J, Li HW, Rossignol PA. 2002. Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology* 83:1372-1385.

Forsyth M, VanLeeuwen J. 1997. Making tradeoffs for agroecosystem health. *Ecosystem Health* 3(2):82-93.

Gardner MR Ashby WJ. 1970. Connectance of large dynamical (cybernetic) systems: critical values for stability. *Nature* 228: 784.

Hall AD, Fagan RE. 1956. Definition of a system. *Gen Sys* 1:18-29.

Householder A, Manion A, Pesante L, Weaver GM. 2001. Managing the threat of denial-of-service attacks. CERT Coordination Center. Carnegie Mellon University.

Jorgensen J, Rossignol AM, Puccia CJ, Levins R, Rossignol PA. 2000. On the variance of eigenvalues of the community matrix: derivation and appraisal. *Ecology* 81: 2928-2931.

King J. 06 October 2001. Suspected terrorists mirror hijackers' actions. CNN.com. /http://www.cnn.com/2001/US/10/06/inv.us.terrorist.threat/index.html

Kumar R, Raghavan P, Rajagopalan S, Tomkins A. 1999. Trawling the web for emerging cyber-communities. In: *Proceedings of the 8th World-Wide Web Conference*.

Levins R. 1968. *Evolution in Changing Environments: Some Theoretical Explorations*. Princeton University Press, Princeton.

May R M. 1972. Will a large complex system be stable? *Nature* 238: 413-414

May RM. 1973. Qualitative stability in model ecosystems. *Ecology* 54: 638–641.

Michener CD, Sokal RR. 1957. A quantitative approach to a problem in classification. *Evolution*, 11:130-162.

Nakajima H. 1992. Sensitivity and stability of flow networks. *Ecological Modelling*. 62:123-133.

O'Neill RV, DeAngelis DL, Waide JB, Allen TFH. 1986. *A Hierarchical Concept of Ecosystems*. Princeton University Press, Princeton, NJ.

Pearl J. 1997. The new challenge: from a century of statistics to an age of causation. UCLA Cognitive Systems Laboratory, Technical Report (R-249), January 1997. Presented at the *IASC Second World Congress*, Pasadena, CA, February 1997. Online: http://singapore.cs.ucla.edu/csl_papers.html

Puccia CJ & Levins R. 1985. *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Harvard University Press, Cambridge, MA.

Purdum TS. How to declare victory. 14 October 2001. *The New York Times*, Week in Review, Section 4, Page 3.

Quirk J. and R Ruppert. 1965. Qualitative economics and the stability of equilibrium. *Review of Economic Studies* 32: 311-326.

Rain Forest Puppy. Full disclosure policy. http://www.wiretrip.net/rfp/policy.html

Reynolds J. 1989. The helminthiasis of the internet. Network Working Group Request for Comments: 1135. http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1135.html

Ricklefs RE. 1990. *Ecology*. WH Freeman and Company, New York.

Schudel G, Wood B, Parks RC. December 6-7, 1999. Modeling behavior of the cyber-terrorist. Position paper for National Security Forum, International Cooperation to Combat Cyber Crime and Terrorism. Hoover Institution, Stanford University, Stanford, CA.

Shelton HH (Approval Authority, Chairman of the Joint Chiefs of Staff). JV2020. US Government Printing Office, Washington DC.

Sneath PH, Sokal RR. 1973. *Numerical Taxonomy: The Principles and Practice of Numerical Classification.* WH Freeman and Company, San Francisco, CA.

Van Regenmortel MH. 1990. Virus species, a much overlooked but essential concept in virus classification. *Intervirology*. 31:241-54.

Walters CJ, Holling CS. 1990. Large-scale management experiments and learning by doing. *Ecology* 7:2060-2068.

Walters CJ. 1986. *Adaptive Management of Renewable Resources*. McGraw-Hill, New York, New York, USA.

Waltner-Toews D, Nielsen O. 1995. Assessing agroecosystem health. Agroecosystem Health Discussion Paper 23, University of Guelph, Guelph, Ontario, Canada.

Wang C, Knight JC, Elder MC. December 2000. On viral infection and the effect of immunization. *16th ACM Annual Computer Applications Conference*, New Orleans, LA. http:\\www.cs.virginia.edu\~jck\recentpapers.htm.

Weaver NC. 2001. A Warhol worm: an internet plague in 15 minutes. http://www.cs.berkeley.edu/~nweaver/warhol.html

Weiss PA (ed). 1971. *Hierarchically Organized Systems in Theory and Practice*. Hafner, New York.

Wimsatt WC. 1997. Aggregativity: Reductive heuristics for finding emergence. *Philosophy of Science*; 64: S372-S384.

APPENDIX A – Ecologist's View of the Insider Threat

In this Appendix we briefly look at the insider threat from an ecological perspective.

The complex nature of critical information systems makes them vulnerable to attack and insider activity poses a significant threat to these systems. Insiders represent one facet of dynamic, interconnected systems of humans and machines. From an abstract perspective, complex systems pose a difficult analytical problem. Given that epidemiologists and ecologists have studiously explored complex biological systems and developed a powerful set of analytical models and tools, we believe it will be informative to explore the parallels between the biological and cyber domains. In this paper, we explore a multidisciplinary approach to the problem based on the human disease control paradigm and examine ecological perspectives to the insider threat.

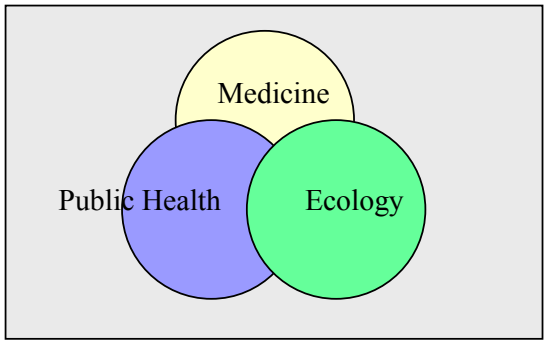
A.1 Multidisciplinary approach for human disease control

An insider has been defined as "any authorized user who performs unauthorized actions. Examples include users, privileged users, system administrators, network administrators, facility support personnel, temporary employees, and contractors." Insider threat has been defined as "any authorized user who performs unauthorized actions that result in loss of control of computational assets."[10]
Challenges to human health are addressed by the interleaved disciplines of medicine, public health, and ecology (Figure A-1). Each addresses a specific level in the hierarchy of analysis: individuals (medicine); human populations (public health); and human populations, disease organisms, and their environment(s) (ecology):

- Medicine is concerned with diagnosis and treatment of *individuals* (cure).
- Public health is concerned with prevention and *population* level diagnosis (prevention).
- Ecology is concerned with tolerance and mitigation (resilience, survival of *systems*).

[10] Anderson, Robert H., Thomas Bozek, Tom Longstaff, Wayne Meitzler, Michael Skroch and Ken Van Wyk. Research on mitigating the insider threat to information systems - #2: Proceedings of a Workshop held August, 2000. http://www.rand.org/publications/CF/CF163 (accessed 12/4/00).

Figure A-1: Relationships among medicine, public health, and ecology



These disciplines also vary with respect to the concept of acceptable loss. Medicine does not conceptually trade off loss to individuals for the gain of others, and maximal measures are taken to treat all individuals. As focus shifts to public health, the health of the population is often acquired at a cost to some individuals. Triage is an example in which an order of treatment based on anticipated benefit to the population (originally of armed combatants) is imposed in large-scale emergencies.

Ecology inherently considers the loss of individuals to maintain the stability of the system. In natural ecosystems, prey are consumed by predators, and plants by herbivores, to persist. The disciplines share a common goal of reducing morbidity and mortality in the human population. The effects of each discipline and their intersections are summarized in Table A-1.

Teasing apart a problem from many perspectives presents alternative approaches for control. Questions posed and hypotheses generated from the medical, public health, and ecological perspectives can be applied to IA as well and are suggested in Table A-2. Control can be initiated at any level. Comprehensive control involves activity at all levels simultaneously.

Table A-1. A multidisciplinary approach to the control of malaria

Human health	Function (Questions addressed)	Example
Medicine	Diagnosis and treatment (What disease is present and how can it be treated?)	Malaria can be diagnosed by blood tests and treated with chemotherapeutic agents.
Intersection of medicine and public health	Risk reduction for individuals (how can the risk of contracting the disease be reduced for individuals?)	Risk to individuals can be reduced by behavioral modifications that will reduce contact with infected mosquitoes.
Public health	Prevention and management of disease (What population-level measures can be taken to reduce the occurrence of disease?)	Large-scale measures such as insecticide spraying will reduce exposure for human populations.
Intersection	Understanding and modifying	Modifying the environment by

of public health and ecology	environmental factors that promote disease (Are there measures that will reduce the risk of disease for populations?)	draining standing water will prevent mosquitoes from breeding.
Ecology	Understanding structure of communities and how they integrate to respond to disease and disease-bearing organisms (How does the system structure support continuance of the disease?)	Human and mosquito populations share a need for clean, accessible water.
Intersection of ecology and medicine	Diagnosis of disease, incorporating environmental assessment (What organisms were present when disease was contracted?)	Malaria is transmitted to humans by a specific genus of mosquitoes (*Anopheles*) and reproduces in humans through a complex biological process.

Table A-2. A multidisciplinary approach to the insider threat

Cyberhealth	Questions addressed	Hypotheses
Medicine	How can the insiders be detected and thwarted?	Insiders are a disease of the system and can be diagnosed using detectors and treated by removal.
Public health	What measures can be taken to ensure system persistence?	Insider threat can be minimized by sacrificing individuals in an affected group.
Ecology	What is the effect of the insider threat on the system-at-large and how can the system-level effect be managed?	Insiders can be managed and regulated.

A.2 Ecological perspective of the insider threat

Ecosystems are described in terms of the *environment* in which they occur, the *resources* they utilize, and the *organisms* that comprise them. A group of interacting, interdependent organisms is called a *community*.

A.3 Resources and environment

Resources are obtained from the environment. There are many potential resources available to an insider. By definition, an insider is 'authorized' and therefore maintained

in the system. Legitimate data, CPU cycles, memory and access to other locations may all be used in an unauthorized manner.

Morris and Rossignol[11] have developed a mathematical theory of interactions that both identifies the types of relationships possible between a consumer and its resource, as well as the likely path of change that can occur from one type to another. Based on parameters characterizing growth of populations, consumer strategies may be regarded as tradeoffs between these parameters. Furthermore, by considering the 'intimacy' of relationships, these tradeoffs gain a degree of complexity that matches our intuition of the natural world.

Pianka[12] described environments as being resource-poor and resource-rich and suggested the types of organisms that might be found in each. Organisms in resource-poor environments typically cannot derive sufficient nutrition unless they consume many organisms. Sheep, for example, are grazers who consume multiple blades of grass. Their relationship with the grass is nonlethal, that is, the individuals of grass need not die, although some may. Lions, on the other hand, also consume multiple prey organisms, but *must* kill them. Their relationship with their prey organisms is lethal.

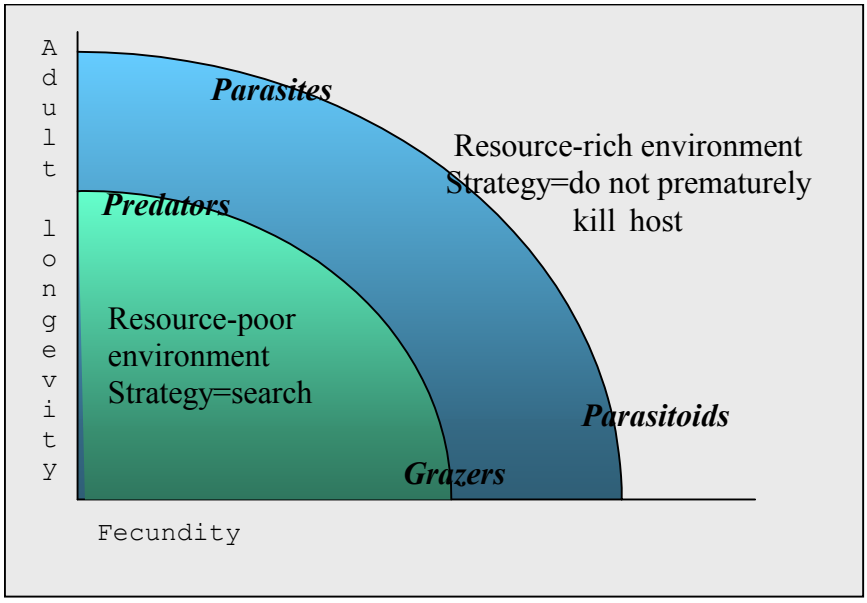
In resource-rich environments, organisms may derive more than adequate nutrition close by. Organisms in such environments, parasites and parasitoids, do not need to search constantly for food. Parasites feed on their hosts in a nonlethal manner. Parasitoids, a more obscure group to nonbiologists, kill their hosts after an appropriate time interval that allows maturation of their young. Predators and parasites enjoy long adult longevity, but do not reproduce at a high rate. Grazers and parasitoids are characterized by shorter life spans, but are highly fecund. The adult stage of parasitoids is very short, barely long enough to mate and infect new hosts with their offspring.

We propose that insiders exist in a resource-rich environment in the organizations in which they are employed or otherwise engaged. Their hosts are the networks in and on which they work. An insider engaged in pilferage of data acts as a parasite, presenting a nonlethal threat to the network. An insider planting a 'logic bomb' acts as a parasitoid whose host may die as a result of his actions after an appropriate interval.

[11] Morris AK and Rossignol PA. Trophic Evolutionary Pathways: A Model Based on Life History Parameters, submitted.

[12] Pianka, E.R. (1970) On r- and K-selection. American Naturalist 104: 592-597.

Figure A-2: Organisms classified by trophic strategies in resource-rich and resource-limited environments



In terms of insider threat, whether or not the insider resembles a parasite or a parasitoid depends on his or her intent. Parasitoids present an acute problem that must be quickly resolved to avoid catastrophic damage. Parasites represent a more chronic problem that can be debilitating, but not lethal in the immediate future.

A.3.1 Models

In the course of his or her activities, an insider may also interact with other people and organizations, any of which may be included in the model of an insider community.

Ecological models are hierarchical in form. Simple ecological models can be combined to form more complex models describing higher level functions. Ecosystem models can address the following levels of analysis of the insider threat:

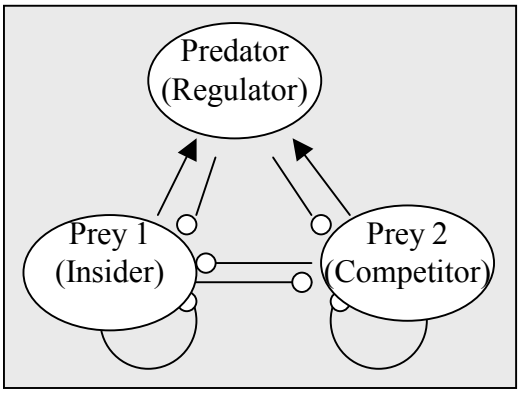
- Thermodynamic model: Where does the insider obtain sustenance?
- Cybernetic model: Can insider threat be internally regulated by the system?
- Evolutionary model: Can we anticipate how insider behavior will change and adapt?

Thermodynamic (trophic or food web) models capture consumption patterns (who eats whom?). They reflect the thermodynamic cascade of energy through the ecosystem.

Cybernetic models incorporate feedback and provide a window into the vulnerabilities of a system. Vulnerabilities to input can be assessed by analysis of the effect of input into each community member. Internal vulnerabilities can be assessed by analysis of the feedback relationships that occur within the community. Cooperation and competition are introduced in cybernetic models. Both can be stabilizing. So-called keystone predators stabilize ecosystems by regulating competing prey populations. For

example, consider two competing populations, elk and deer. The success of one population reduces and potentially eliminates the other. Predators can regulate and stabilize the system by regulating both populations at sustainable levels (Figure A-3). The tradeoff with respect to the insider threat is that the presence of the insider is preserved as a consequence of the stability of the system. Situations in which such a tradeoff might be acceptable include instances where personnel with technical skills who engage in low-level threats, such as gaming, are difficult to recruit and retain. Stability comes with a cost; it is necessary to allocate resources to regulatory links.

Figure A-3. Insider regulated by a keystone predator.[13]

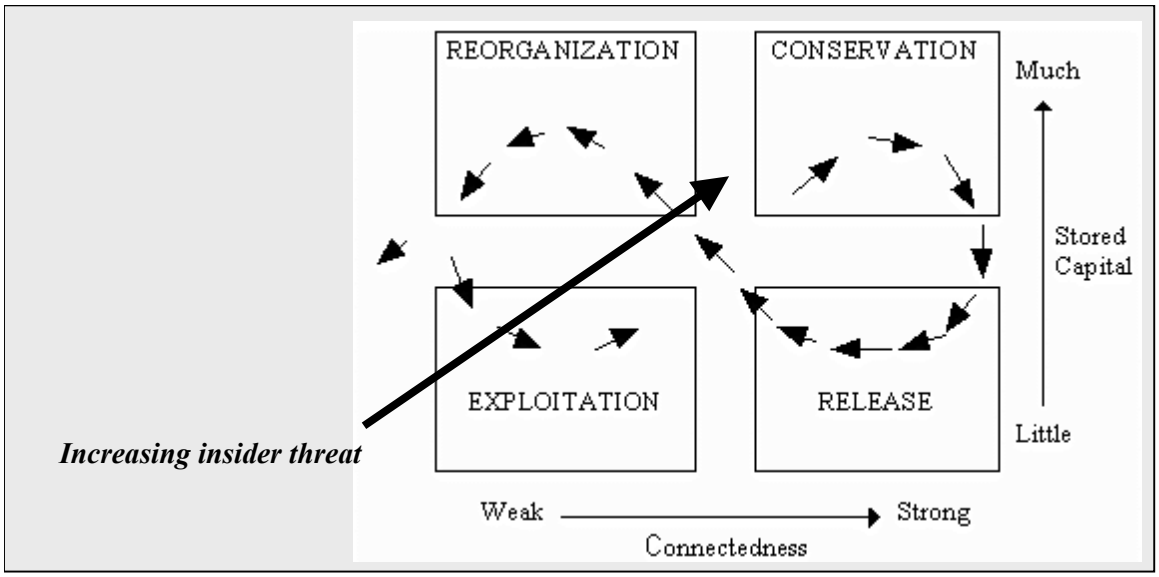


Succession models also suggest models of the effects of insider activity within a system. From the point of view of succession, ecosystems proceed through stages of development. Beginning with a loose aggregation of relatively independent entities, the system becomes more interconnected as it grows. Accordingly, insider activity will be less of a threat to very young organizations. As the organizations grow in size and accommodate more individuals within them (potential insiders), the threat of catastrophic surprise becomes greater. Figure A-4 shows a graphical representation developed by Holling[14], in which the cycle of succession is shown in terms of connectedness and stored capital. Loosely connected species come into physical proximity during the exploitation phase. Resources are plentiful. As communities form, species become more interconnected and more resources are stored within the system, for example, as forests. Release occurs when the increasingly fragile, but extremely stable system represented at the cusp of the curve in the conservation phase is perturbed from equilibrium. Resources are released from stored capital and returned into the environment. The cycle continues when loosely organized survivors or new species recolonize the environment.

[13] This diagram is a signed-digraph of the relationships among three species. Positive effects, those that cause an increase to one species from another, are represented with arrows. Negative effects, those that cause a decrease to a species are represented with lines terminated in bubbles.

[14] *(in Janssen, M. A. and S. R. Carpenter. 1999. Managing the Resilience of Lakes: A multi-agent modeling approach. Conservation Ecology 3(2): 15. [online] URL: http://www.consecol.org/vol3/iss2/art15)*

Figure A-4. Holling's succession model



Given a stable system that includes insiders, evolutionary models suggest how insiders will change as technology advances. Metaphors to fitness may provide an understanding of how ecologically important traits may express themselves among insiders as a group in the future. For example, social engineering skills, such as the ability to convince an operator to verbally divulge passwords, may provide an advantage for insiders with this skill. Through time, this skill will become more common as those who possess it are more successful.

Success among insiders differs from success among biological organisms because it does not involve reproductive potential, in the sense of offspring. However, alternate pathways for insiders to obtain, accumulate, and disseminate tools exist. For example, mature hackers do produce more hackers in a didactically reproductive process.

A.4 Data

Ecological communities are described in terms of direct and indirect effects. Direct effects express the direct, one-to-one impact of one community member upon another. Indirect effects express the impact of one community member upon another through feedback. Potential sources of direct data include event logs and reports from filters. Indirect data might be obtained by tracking data pedigree.

A.5 Analysis

Model formulation is based on observation of population levels and interactions among organisms. These models inform hypotheses about ecosystem structure.

Like many computer networks, ecosystems are unique. Natural experiments, such as floods, fires, and el Niño, cannot be replicated and are observational. Controlled experiments are conducted on small sections of ecosystems. For example, transects on the sea floor may be enclosed in cages to prevent the entry of predators, and the effects

on organisms within transects may be observed. Structural information about computer networks might also be obtained from such press experiments. For example, to assess the level of insider threat, so-called 'press' experiments (a permanent change in a growth parameter of a population) might consist of diversions in which alternate data sources were made available and access to them observed. Mathematical analysis (in this case, the inverse of the community matrix) provides a prediction against which results can be interpreted. These analyses are discussed in detail in Chapter 5.

A.6 Conclusion

The insider threat cannot be addressed in isolation. Observations from many levels are necessary to form broad, inclusive models spanning detection, prevention, and tolerance. The ecological perspective of the insider threat suggests hypotheses such as the following:

- Insider activity can be regulated with competition.
- Insider activity is less damaging in loosely organized systems.
- Environmental modifications (press experiments) allow observation of system structure incorporating insider activity.

The insider threat is more than a technological problem and will require more than a technological solution. The health-ecology paradigm provides a template on which to base models of management and control and provides proven analytic techniques for assessment.

Appendix B - Brief Review of Computer Taxonomies

In this Appendix, we briefly review current taxonomies for computer attacks and malicious code.

B.1 Classification of attacks

Taxonomies of malicious computer attacks have been constructed, using computer-appropriate characters. Howard evaluated examples of taxonomic scheme against characteristics of satisfactory taxonomies presented by Amoroso (1994). According to these criteria, taxonomies should have classification criteria that are

- Mutually exclusive
- Exhaustive
- Unambiguous
- Repeatable
- Accepted and useful

Howard has described a broad range of taxonomies and proposed his own comprehensive scheme. In general order of increasing complexity, existing taxonomies of computer and network attacks rely on the following methods:

- Lists of terms
- Lists of categories
- Results categories
- Empirical lists
- Matrices

He found each of the above methods to be flawed based on one or more of these criteria. His findings are shown in Table B-1. Please see Howard (1997) for a more complete discussion.

Table B-1. Howard's taxonomic scheme

Taxonomic method	Example (reference)	Unmet criteria
Lists of terms	Login spoofing/Induced stress failures/Network services attacks/Combined attacks (Cohen 95:40-54)	Not mutually exclusive Unmanageably long Difficult to apply No structure
Lists of categories	Stealing passwords Social engineering … Denial of service (Cheswick and Bellovin, 1994)	Not mutually exclusive Unmanageably long Difficult to apply
Results categories (describe result of attack)	Corruption Leakage Denial (Cohen 95:55)	Not exclusive
Empirical lists (based on classification of empirical data)	External information theft External abuse of resources . . . (Neumann and Parker, 1989)	Not logical or intuitive
Matrices (two dimensional classification schemes)	Security flaw taxonomy: Flaws by genesis (Landwehr, et al, 1994)	Not unambiguous Not exhaustive

B.2 Operational models

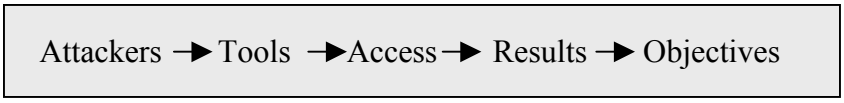
Stallings (1995) developed a process model of security threats in which he defined passive and active attacks:

Table B-2. Stallings' model of security threat

Attack type	Category of attack	Description
Passive	Interception	Unauthorized party gains access to an asset
Active	Interruption	Asset is destroyed, becomes unavailable or unusable
	Modification	Unauthorized party gains access to and tampers with an asset
	Fabrication	Unauthorized party inserts counterfeit objects into the system

Howard (1997) suggested a process model linking attackers to their objectives through tools, access and results.

Figure B-1. Howard's model



Attackers include hackers, spies, terrorists, corporate raiders, professional criminals and vandals. Examples of tools include user commands, scripts or programs, autonomous agents, toolkits, distributed tools and data taps. Access is defined as a vulnerability leading to unauthorized access or use which processes files or data. The results of the attack can be corruption of information, disclosure of information, theft of service or denial-of-service. The objectives of the attack may be to achieve political or financial gain, or to cause damage.

B.3 Malicious code taxonomies

B.3.1 Functional descriptions

The most familiar taxonomic descriptions are for malicious code:

- Hoax – usually dispersed as a chain letter by email
- Trojan Horse – a program that neither replicates nor copies itself, but inflicts damage or compromises security
- Virus – A program or code that replicates
- Worm – A program that makes copies of itself.

An extensive taxonomy has been developed for viruses.[15] They can be classified according to what they infect.

- System sector viruses – infect system files
- File viruses – infect program (.com and .exe) files
- Macro viruses – infect data files
- Companion viruses – add files that run first to disk
- Cluster viruses – Infect through the disk directory
- Batch file viruses – Infect using text batch files
- Source code viruses – Add code to program source code

Viruses can also be classified according to how they infect:

- Polymorphic viruses – change characteristics as they infect
- Stealth viruses – actively hide from anti-virus or system hardware
- Fast and slow infectors – infect in a particular way to avoid anti-virus software
- Sparse viruses – infect infrequently

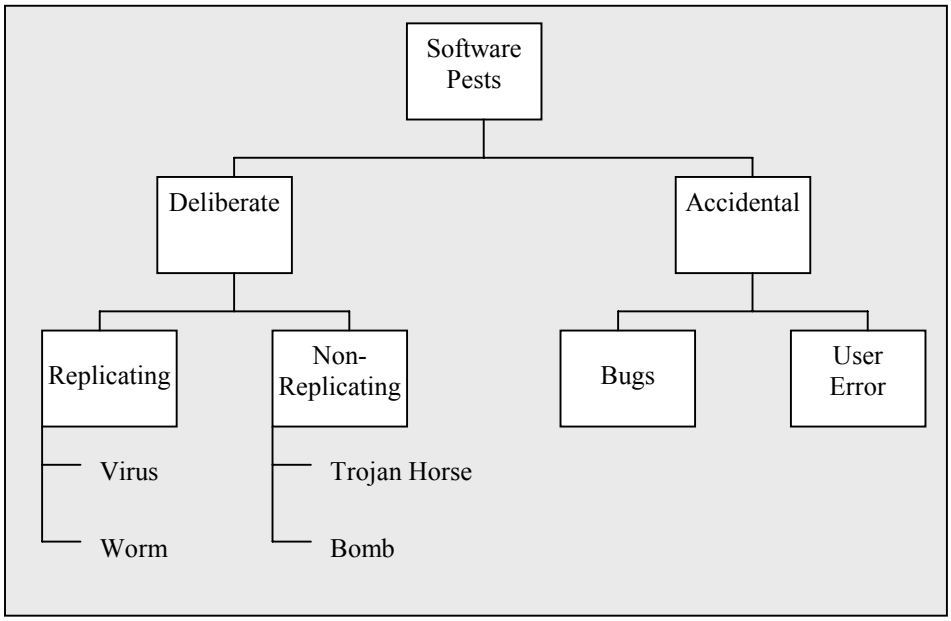
[15] http://www.mcafee.com/anti-virus/virus_glossary.asp#top and http://www.cknow.com/vtutor/vtarmored_m.html

- Armored viruses – programmed to make disassembly difficult
- Multipartite viruses – fall into multiple categories
- Cavity viruses – attempt to maintain a constant size while infecting
- Tunneling viruses – attempt to "tunnel under" anti-virus software while infecting
- Camouflage viruses – attempt to appear as a benign program to scanners
- NTFS ADS viruses – ride on alternate data streams in the NT file system

B.4 Hierarchical descriptions

Louw and Duffy (1992) presented a hierarchical depiction of software pests, but this scheme is so general that it is severely limited as a taxonomy (Figure B-2).

Figure B- 2. Louw and Duffy's hierarchy of software pests



B.5 Outcome descriptions

SARC includes a threat severity assessment for malicious code. Metrics include:

- Wild – extent to which a virus is spreading (range)
- Damage – potential damage that an infection could inflict
- Distribution – how quickly a program spreads itself.

These metrics are combined into an overall severity measure ranging from Category 1 (minimal) to Category 5 (Very Severe)

Adleman (1988) constructed a taxonomy of viruses using the dimensions of pathogenicity (producing injury) and contagiousness (ability to spread). He classified viruses into four disjoint (independent) and mutually exclusive categories:

- Benign
- Epeian (after the builder of the original Trojan horse of the Odyssey)
- Disseminating
- Malicious

The relative contagiousness and pathogenicity of these agents is shown in Table B-3.

Table B-3. Contagiousness and pathogenicity of viruses in Adleman's model

	Pathogenicity	
Contagiousness	Low	High
Low	Benign	Epeian
High	Disseminating	Malicious

Adleman also postulated paths of infection based upon these categories:

- programs infected by a benign virus will be benignant with respect to their uninfected predecessors;
- programs infected by an Epeian virus can only be benignant of Trojan horses with respect to their uninfected predecessors; *i.e.*, they will not be able to spread themselves;
- programs infected by a disseminating virus can only be benignant of carriers with respect to their uninfected predecessors; *i.e.*, they are never pathogenic.

Adleman did not explore complex attacks composed of multiple agents.

B.6 References

Adleman LM. 1988. An abstract theory of computer viruses. In: Goos G, Hartmanis J, eds. *Advances in Cryptology—Crypto '88*. Springer-Verlag, New York

Amoroso EG. 1994. *Fundamentals of Computer Security Technology*. Prentice-Hall PTR, Upper Saddle River, NJ. Cited In: Howard JD. 1997. *An Analysis of Security Incidents on the Internet: 1989-1995*. Thesis Dissertation for Carnegie Mellon University. Pittsburgh, PA. http://www.cert.org:80/research/JHThesis/Start.html. (accessed 12/26/00)

Cheswick WR, Bellovin SM. 1994. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley, Reading, MA. Cited In: Howard JD. 1997. *An Analysis of Security Incidents on the Internet: 1989-1995*. Thesis Dissertation for Carnegie Mellon University. Pittsburgh, PA. http://www.cert.org:80/research/JHThesis/Start.html. (accessed 12/26/00)

Cohen FB. 1995. *Protection and Security on the Information Superhighway*. John Wiley and Sons, New York. Cited In: Howard JD. 1997. *An Analysis of Security Incidents on the Internet: 1989-1995*. Thesis Dissertation for Carnegie Mellon University. Pittsburgh, PA. http://www.cert.org:80/research/JHThesis/Start.html. (accessed 12/26/00)

Howard JD. 1997. *An Analysis of Security Incidents on the Internet: 1989-1995*. Thesis Dissertation for Carnegie Mellon University. Pittsburgh, PA. http://www.cert.org:80/research/JHThesis/Start.html. (accessed 12/26/00)

Landwehr CE, Bull AR, McDermott JP, Choi WS. 1994. "A Taxonomy of Computer Security Flaws." *ACM Computing Surveys*. 26(3): 211-254. Cited In: Howard JD. 1997. *An Analysis of Security Incidents on the Internet: 1989-1995*. Thesis Dissertation for

Carnegie Mellon University. Pittsburgh, PA.
http://www.cert.org:80/research/JHThesis/Start.html. (accessed 12/26/00)

Louw E, Duffy N. 1992. *Managing Computer Viruses*. Oxford University Press. Oxford.

Neumann P, Parker D. 1989. A summary of computer misuse techniques." *Proceedings of the 12th National Computer Security Conference*. Cited In: Howard JD. 1997. *An Analysis of Security Incidents on the Internet: 1989-1995*. Thesis Dissertation for Carnegie Mellon University. Pittsburgh, PA.
http://www.cert.org:80/research/JHThesis/Start.html. (accessed 12/26/00)

APPENDIX C - Epidemiology Examples

C.1 Introduction

This Appendix contains details about the calculation of basic reproduction rate, generation time, and doubling time for three email viruses (PrettyPark, LoveLetter, and Anna), two worms (MTX and Kak), and one macrovirus (Ethan). Graphs describing their spread, obtained from data provided by two virus protection companies, are also given for each virus or worm.

The purpose of these examples is to illustrate the process through which epidemiological measurements may be made for malicious code. The parameters used in the calculations are best-guess estimates.

Each example contains

- a technical description presented in a two-column format. The left column of each description contains information taken from a published mechanical description obtained from an anti-virus company website. Sections of the technical description that are used in the calculations are highlighted. Our annotations describing important points in the calculation of the BRR appear in the right column adjacent to this text;
- a life cycle diagram that calls out the information from the mechanical description that applies to transmission;
- calculation of basic reproduction rate (BRR). When there are multiple methods of transmission (for example, email, mIRC, or in file, a BRR is given for each branch individually, and for the total of all branches (the sum of individual branch BRRs);
- generation time calculated as the sum of delays that occur during transmission between executable forms of the virus or, for worms that continue to infect over a relatively long period of time, as the lifetime of the infection;
- doubling time calculated from the formula: Doubling Time = Generation Time / $\log_2$ BRR;
- graphs of data obtained from two anti-virus companies.

When information is available for any one virus or worm from the two anti-virus companies, stark differences between the data may be apparent. This is due to the differences in the method of data collection used by the companies. Company 1 filters emails at scanning towers, where they are scanned and relayed to recipients. The data from this company resemble true incidence. The data from Company 2 is obtained when customers initiate virus scans.

C.2 Pretty Park

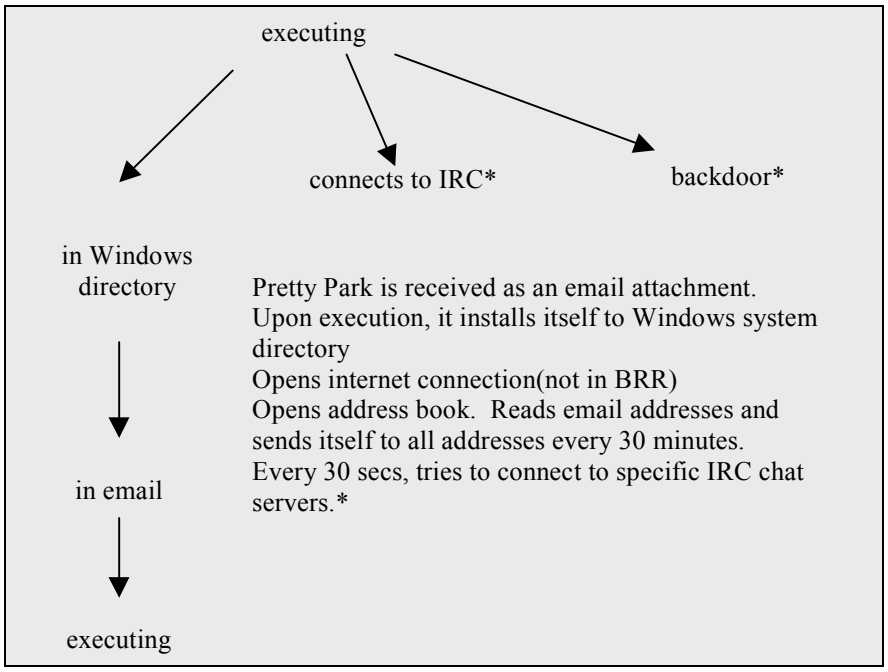
C.2.1 Technical description

The following technical description was published by F-Secure Corporation at http://www.europe.f-secure.com/prettyp.shtml.

NAME: PrettyPark ALIAS: Pretty Park, I-Worm.PrettyPark The 'PrettyPark' also known as 'Trojan.PSW.CHV' is an Internet worm, a password stealing trojan and a backdoor at the same time. It was reported to be widespread in Central Europe in June 1999. There was also an outbreak of this worm in March 2000. Several variants of Pretty Park are known. All of them have the same functionality, but some are packed.	ANNOTATIONS
PrettyPark spreads itself via Internet by attaching its body to e-mails as 'Pretty Park.Exe' file. The file has the icon showing a character or the famous cartoon serial called South Park. Being executed it installs itself to system and then sends e-mail messages with its copy attached to addresses listed in Address Book and also informs someone (most likely worm author) on specific IRC servers about infected system settings and passwords. It also can be used as a backdoor (remote access tool).	PrettyPark begins its execution cycle as an email attachment
When the worm is executed in the system for the first time, it looks for its copy already active in memory. The worm does this by looking for application that has "#32770" window caption. If there is no such window, the worm registers itself as a hidden application (not visible in the task list) and runs its installation routine.	
While installing to system the worm copies itself to \Windows\System\ directory as FILES32.VXD file and then modifies the Registry to be run each time any EXE file starts when Windows is active. The worm does this by modifying an EXE file startup command key in the HKEY_CLASSES_ROOT. The key name is exefile\shell\open\command and it is associated with the worm file (FILES32.VXD file that was created in the Windows system folder). If the FILES32.VXD file is deleted and Registry is not corrected, the EXE files would not start any more.	

In case of error during installing the worm activates the SSPIPES.SCR screen saver (3D Pipes). If this file is missing, the worm tries to activate. 'Canalisation3D.SCR' screen saver.	
Then the worm opens Internet connection and activates 2 its routines. Further on these inits socket (Internet) connection and runs its routines that are activated regularly: the first one once per 30 seconds, another one - once per 30 minutes. The first routine that activates once in 30 seconds tries to connect to one of IRC chat servers (see the list below) and to send a messages to someone if he is present on any channel of this chat server. This allows worm author to monitor infected computers. The list of IRC servers the worm tries to connect to: irc.twiny.net irc.stealth.net irc.grolier.net irc.club-internet.fr ircnet.irc.aol.com irc.emn.fr irc.anet.com irc.insat.com irc.ncal.verio.net irc.cifnet.com irc.skybel.net irc.eurecom.fr irc.easynet.co.uk	IRC branch is not involved in replication and is not included in BRR calculation.
The worm may be also used as a backdoor (remote access tool) by its author. It can send out system configuration details, drives list, directories info as well as confidential information: Internet access passwords and telephone numbers, Remote Access Service login names and passwords, ICQ numbers, etc. The backdoor is also able to create/remove directories, send/receive files, delete and execute them, etc.	
The second routine, which is activated once per 30 minutes, opens Address Book file, reads e-mail addresses from there, and sends messages to these addresses. The message Subject field contains the text: C:\CoolProgs\Pretty Park.exe The message has an attached copy of the worm as Pretty Park.EXE file. If someone receives this message and runs the attached file his system becomes infected.	User must click on email attachment to execute.

Figure C-1. Life cycle of PrettyPark



C.2.2 Life cycle

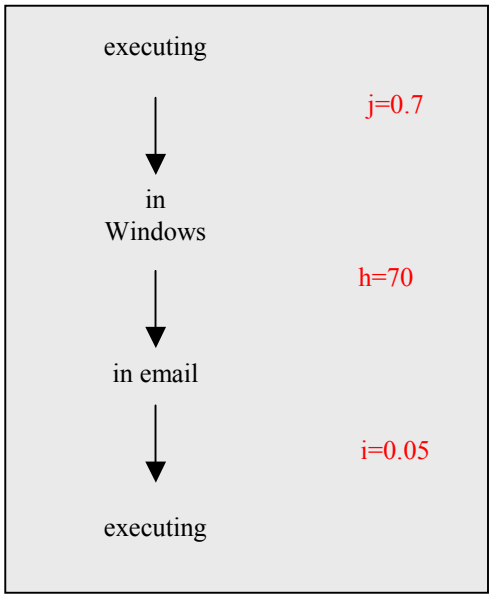
The life cycle for PrettyPark is shown in Figure C-1.

C.2.3 Basic reproduction rate (BRR)

The parameters used in calculation of the BRR for PrettyPark are shown in Figure C-2. The default parameter for the probability that the email attachment will be read has been modified because the worm sends multiple copies to each email address, which is suspicious. The estimated BRR for Pretty Park [j * h * i] is 0.7 * 70 * .05 = 2.45

Parameter description:
h=mean number of addresses in address book
i=P(attachment will be opened); lower than default because topic uninteresting and multiple copies are suspicious
j=P(use Outlook and Windows)

Figure C-2. Basic reproduction rate parameters for Pretty Park

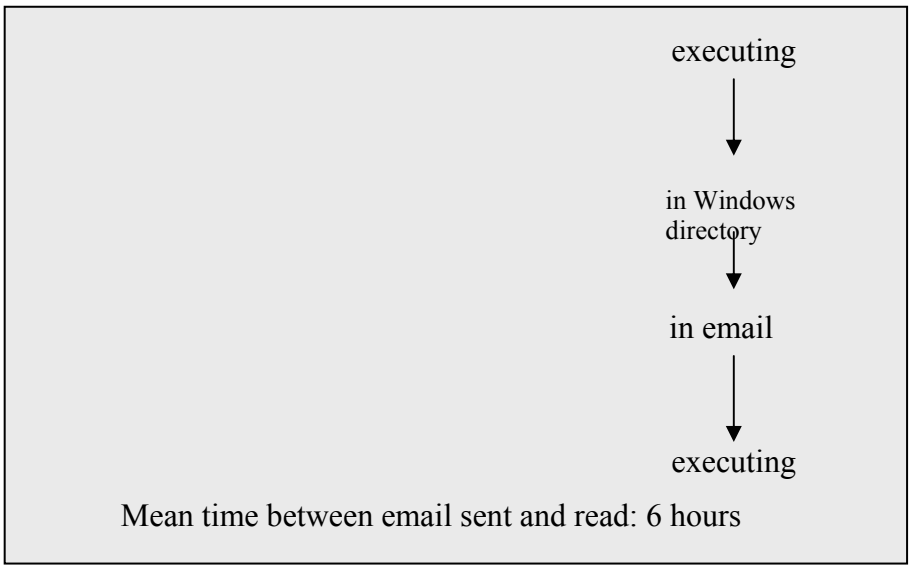


NOTE: Sending every 30 minutes does not impact BRR because it is sending to the same addresses; it does cause collateral damage as a DOS attack

C.2.4 Generation time

The generation time for Pretty Park is estimated to be six hours and is due to one delay shown in Figure C-3.

Figure C-3. Generation time diagram for PrettyPark



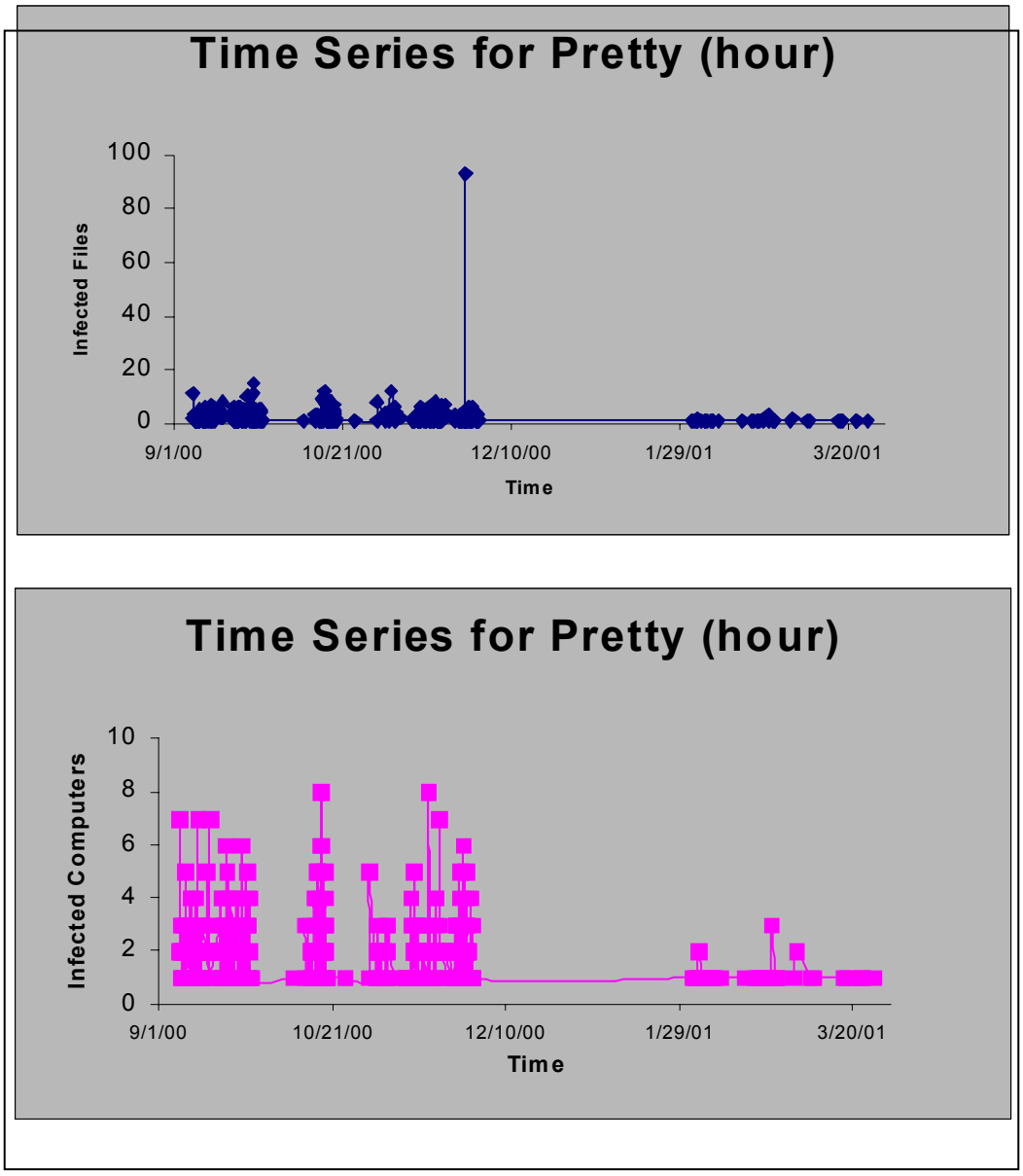
C.2.5 Doubling time

Estimated doubling time for Pretty Park is generation time/ $\log_2$ BRR = 4.6 hours

C.2.6 Anti-virus data

Data illustrating the number or infections recorded by Anti-Virus Company 2 from 01 September 2001 to 20 March 2001 are given in Figure C-4. As discussed in

Figure C-4. Number of infections per hour for PrettyPark reported by Anti-Virus Company 2 (expressed as number of infected files (top) and number of infected machines (bottom))



Section 2, these data do not show clear trends.

C.3 LoveLetter

C.3.1 Technical description

The following technical description was published by F-Secure Corporation at http://www.europe.f-secure.com/v-dscs/love.shtml.

	ANNOTATIONS
NAME: LoveLetter ALIAS: Lovebug, I-Worm.LoveLetter, ILOVEYOU VBS/LoveLetter is a VBScript worm. It spreads through e-mail as a chain letter. You can protect yourself against VBScript worms by uninstalling the Windows Script Host. For further information, please look at http://www.F-Secure.com/virus-info/u-vbs/ VARIANT: LoveLetter.A The worm uses the Outlook e-mail application to spread. LoveLetter is also an overwriting VBS virus and it spreads using a mIRC client as well. When it is executed, it first copies itself to the Windows System directory as: - MSKernel32.vbs - LOVE-LETTER-FOR-YOU.TXT.vbs and to the Windows directory as: - Win32DLL.vbs Then it adds itself to the registry, so that it will be executed when the system is restarted. It adds the following registry keys:	There are three transmission paths to include in the BRR calculation.

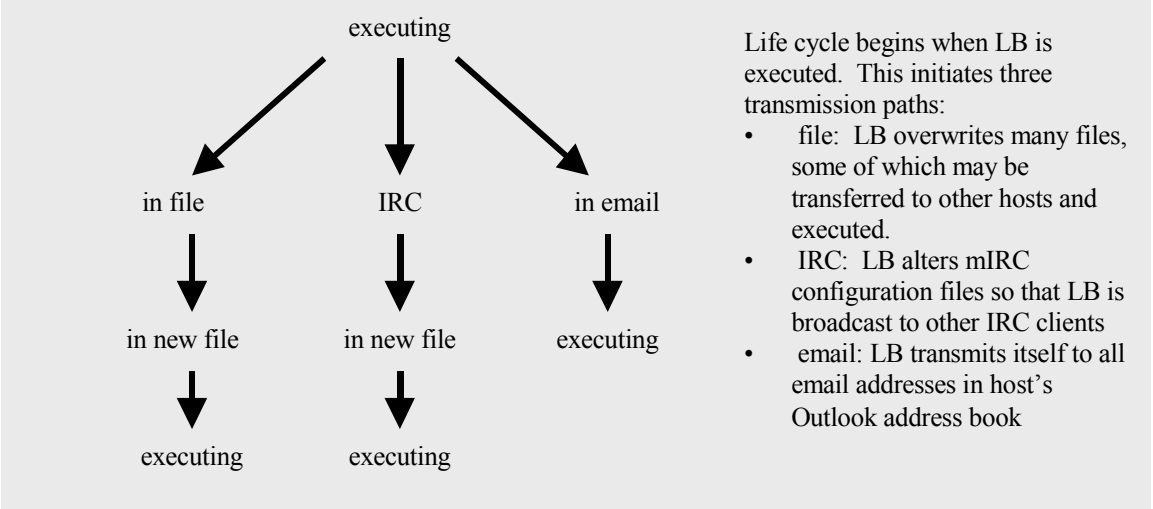
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\MSKernel32 HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices\Win32DLL After that the worm replaces the Internet Explorer home page with a link that points to an executable program, "WIN-BUGSFIX.exe". If the file is downloaded, the worm adds this to the registry as well, which causes the program to be executed when the system is restarted. The executable part the LoveLetter worm downloads from the web is a password stealing trojan. On then system startup the trojan tries to find a hidden window named 'BAROK...'. If it is present, the trojan exits immediately, in other case the main routine takes control. The trojan checks for the WinFAT32 subkey in the following Registry key: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run If the WinFAT32 subkey key is not found, the trojan creates it, copies itself to the \Windows\System\ directory as WINFAT32.EXE and then it runs the file from that location. The above registry key modification causes the trojan to become active every time Windows starts. Then the trojan sets the Internet Explorer startup page to 'about:blank'. After that the trojan tries to find and delete the following keys: Software\Microsoft\Windows\CurrentVersion\Policies\Network\HideSharePwds Software\Microsoft\Windows\CurrentVersi	Reboot is required. Not included in BRR calculation because does not pertain to replication.

on\Policies\Network\DisablePwdCaching .DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network\HideSharePwds .DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network\DisablePwdCaching Then the trojan registers a new window class and creates a hidden window titled 'BAROK...' and remains resident in the Windows memory as a hidden application. Immediately after startup and when timer counters reach certain values, the trojan loads the MPR.DLL library, calls the WNetEnumCashedPasswords function and sends stolen RAS passwords and all cached Windows passwords to e-mail address 'mailme@super.net.ph' that most likely belongs to the trojan's author. The trojan uses mail server 'smtp.super.net.ph' to send e-mails. The e-mail's subject is 'Barok... email.passwords.sender.trojan'. There is the author's copyright message inside the trojan's body: barok ...i hate go to school suck ->by:spyder @Copyright (c) 2000 GRAMMERSoft Group >Manila,Phils. There are also some encrypted text messages in the trojan's body for its own use. After that the worm creates an HTML file called "LOVE-LETTER-FOR-YOU.HTM" to the Windows System directory. This file contains the worm and it will be sent using mIRC whenever another person joins an IRC channel where the infected user currently is. To accomplish this the worm	mIRC transmission path. Emails itself to all addresses in Outlook address book. Only one message sent to each email address.

replaces the "script.ini" file from the mIRC installation directory. Then the worm uses Outlook to mass mail itself to everyone in each address book. The message that it sends looks like this: Subject: ILOVEYOU Body: kindly check the attached LOVELETTER coming from me. Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs LoveLetter sends the mail once to each recipient. After a mail has been sent, it adds a marker to the registry and does not mass mail itself anymore. Then the virus searches for certain file types from all folders in all local and remote drives and overwrites them with its own code. The files that are overwritten have either a "vbs" or a "vbe" extension. The virus creates a new file with the same name for files with the following extensions: ".js", ".jse", ".css", ".wsh", ".sct" and ".hta". The only difference is that the extension of the new file is ".vbs". The original file will be deleted. After this has been done, the the virus locates files with ".jpg" and ".jpeg" extensions, adds a new file next to it and deletes the original file. Then the virus locates ".mp3" and ".mp2" files, creates a new file and hides the original file. In both cases the new files created will have the original name with the additional extension ".vbs". For example, a picture named "pic.jpg" will cause a new file called "pic.jpg.vbs" to be created.	Overwrites files with specific extensions.

C.4 Life cycle

Figure C-5. Life cycle of LoveLetter (LB)

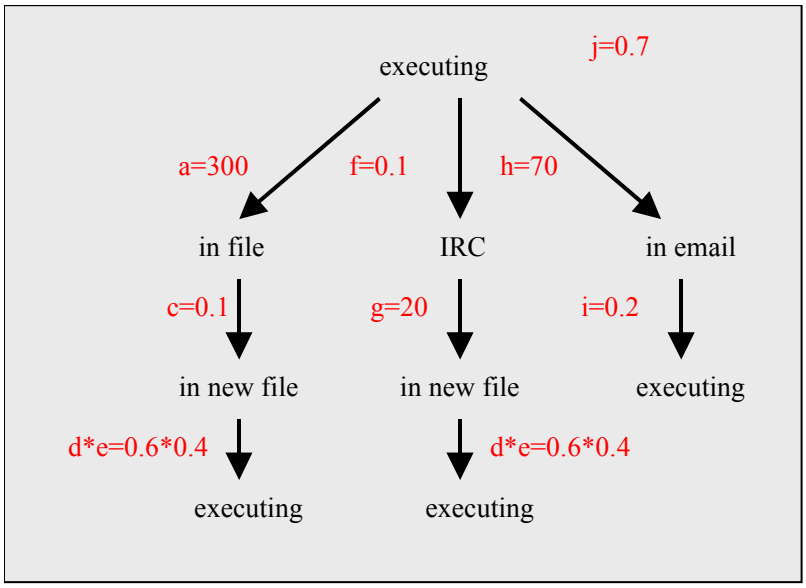


The life cycle of LoveLetter is shown in Figure C-5.

C.4.1 Basic reproduction rate (BRR)

The parameters used in the calculation of the BRR for LoveLetter are shown in Figure C-6. LoveLetter was very successful because it contained an attractive message from a trusted source. To reflect this, we have increased the probability that an email attachment will be executed to 0.2.

Figure C-6. Basic reproduction rate parameters for LoveLetter



Parameter description:
a=mean number of files overwritten
c=P(file being transferred to another machine)
d=P(recipient does not delete file on arrival)
e=P(recipient runs file given it was not deleted)
f=P(mIRC is used on a machine)
g=mean number of IRC users on a channel at a given time
h=mean number of addresses in address book
i=P(attachment will be opened)
j=P(use Outlook and Windows)

The estimated BRR for LoveLetter is given for each of the three paths in the analytical model. The total BRR is the sum of these components.

BRR total = BRR (in file) + BRR (IRC) + BRR (email)

= acde + fgde + hi

= 5.04 + 0.34 + 9.8 = 15.18

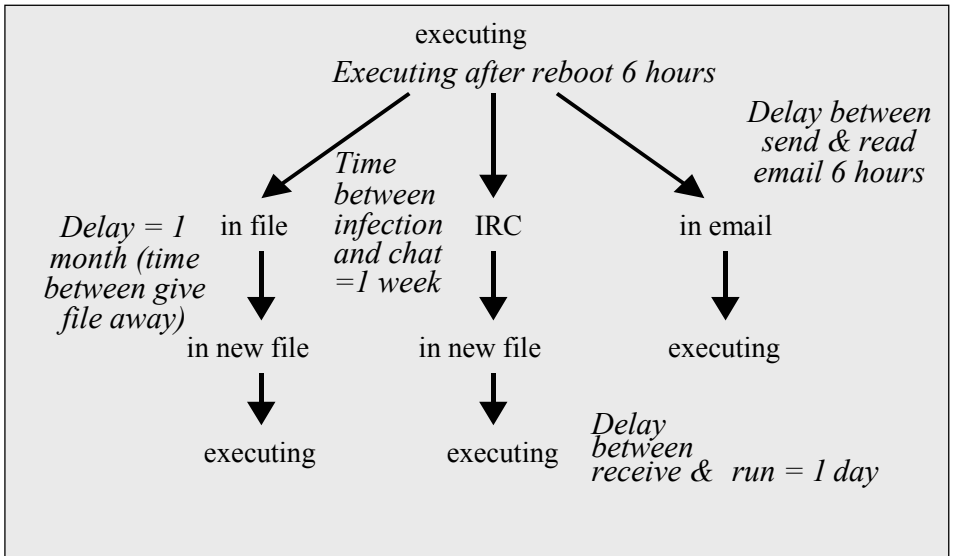
C.4.2 Generation time

The estimated generation time for LoveLetter is illustrated in Figure C-7. We calculate the generation time for each transmission branch independently:

Generation time (in file) = one month + six hours
Generation time (IRC) = eight days + six hours
Generation time (email) = twelve hours

Examining the BRR in conjunction with generation time gives an indication of the relative contributions of each transmission path to infectious spread. BRR (email) is high and generation time is short. This branch is responsible for the initial epidemic spike. BRR (in file) is high, but the generation time is very long. The infection is slow to spread to this transmission path. This path may contribute to the persistence of the infection in the population after the initial epidemic rise has subsided. BRR (IRC) is slight and the generation time is relatively long when compared to the generation time for BRR (email). This transmission path contributes little to infectious spread in this model.

Figure C-7. Generation time diagram for LoveLetter



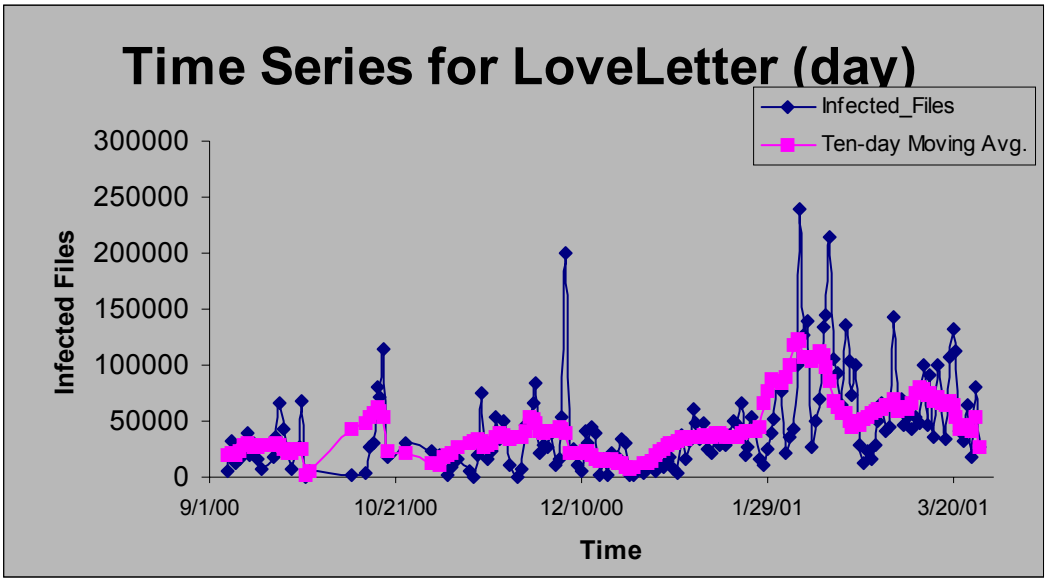
C.4.3 Doubling time

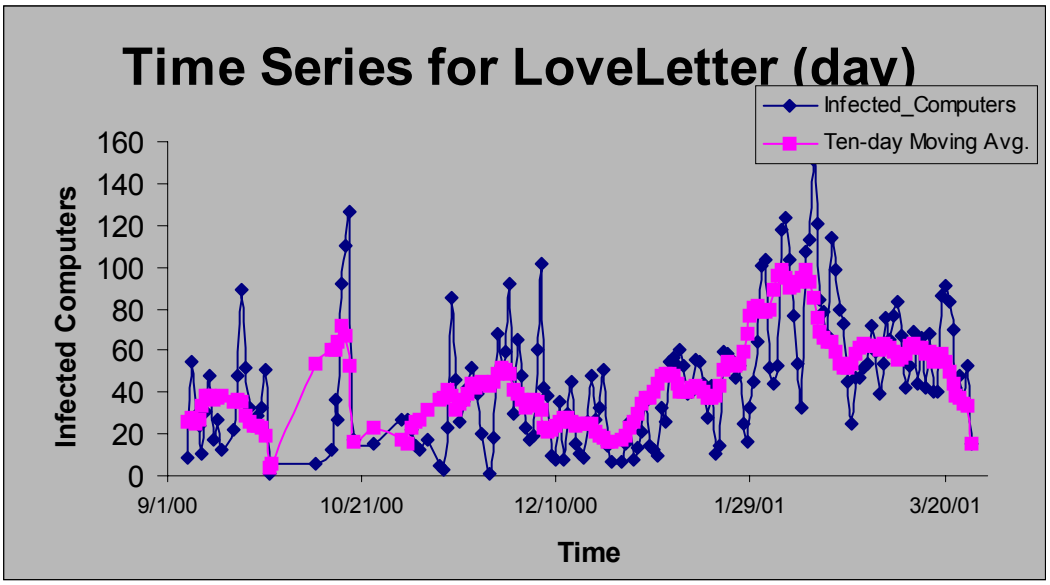
The estimated doubling time for the email form of LoveLetter is 3.6 hours. Published reports estimated the total number of opened attachments at 1.9 million in one day. Using the calculations in our model, a total of 1.9 million infections would be achieved along the email transmission branch between 7 and 8 generations (25.2 – 28.8 hours).

C.4.4 Anti-virus data

We were not able to obtain real-world data that captured the initial epidemic spread of the LoveLetter worm. The data in Figure C-8 show the subsequent level of persistence in the population. The release date for LoveLetter was 04 May 2000.

Figure C-8. Number of infections per day for LoveLetter Reported by Anti-Virus Company 2 (expressed as number of infected files (top) and number of infected machines (bottom))





C.5 Anna

C.5.1 Technical description

The following technical description was published by Greenapple.com at http://greenapple.com/support/security/pc-sentry/sst.htm

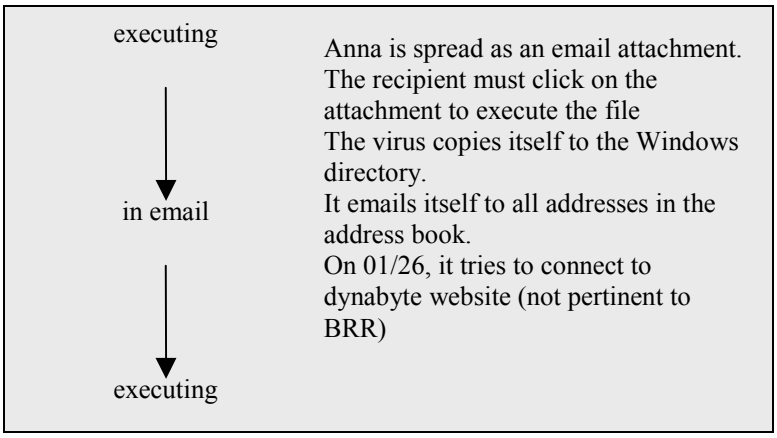
name: VBS.SST@mm aka: AnnKournikova.jpg, On The Fly, Kalamar type: Worm host platform: Windows first incidence: None yet last incidence: None yet level of incidence: Low damage capacity: Low links: McAfee, Norton look for: E-mail with attachments claiming to contain pictures of the tennis player Anna Kournikova. The file attached is the virus. Keep an eye out for subjects such as "Here you are ;-)," "Here you have ;o)" and "Here you go ;-)."	ANNOTATIONS
A recently discovered virus that spreads like the "ILOVEYOU" virus of the past, VBS.SST uses an email with an attachment claiming to be an image of the tennis player Anna Kournikova. This is no picture, but a Visual Basic script file which, when opened, infects the machine. As part of infections, the virus will mail itself out to others in the address book of the user.	Spreads via an email attachment. Automatically emails itself to addresses in address book.
The following symptoms indicate a probable infection by the virus: : Existence of the file "c:\WINDOWS\AnnaKournikova.jpg.vbs" : Existence of the registry key HKEY_USERS\.DEFAULT\Software\OnTheFly	There is no mention of a need to reboot.
To remove the virus download the latest engine and 'dat' files for your virus detection software and run a virus scan.	

Machines that have Microsoft's "Scriptlet.eyedog" patch installed (a patch which helps contain renegade VBS scripts) should be ok. Windows 95, 98, Me and 2000 machines that do not have Windows Scripting installed will be safe. Most machine, however, do have Scripting installed and users should download the latest engine and 'dat' files for their virus software.	

C.5.2 Life cycle

The conceptualized life cycle for Anna is shown in Figure C-9.

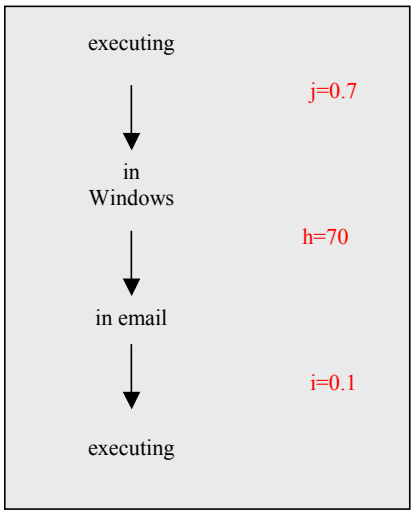
Figure C-9. Life cycle of Anna



C.5.3 Basic reproduction rate (BRR)

The parameters used in estimating the BRR of Anna are shown in Figure C-10.

Figure C-10. Basic reproduction rate parameters for Anna



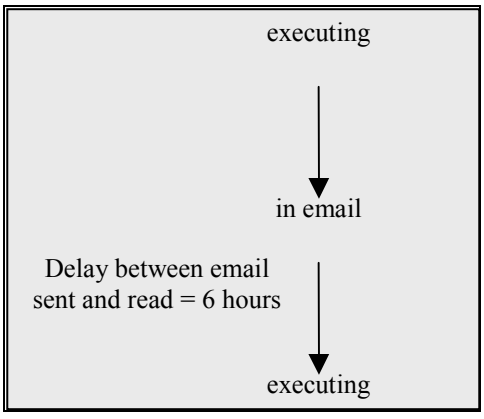
Parameter description:
h=mean number of addresses in address book
i=P(attachment will be opened)
j=P(use Outlook and Windows)

The estimated BRR for Anna is 0.7 * 70*0.1 = 4.9

C.5.4 Generation time

The estimated generation time for Anna is 6 hours as illustrated in Figure C-11.

Figure C-11. Generation time diagram for Anna



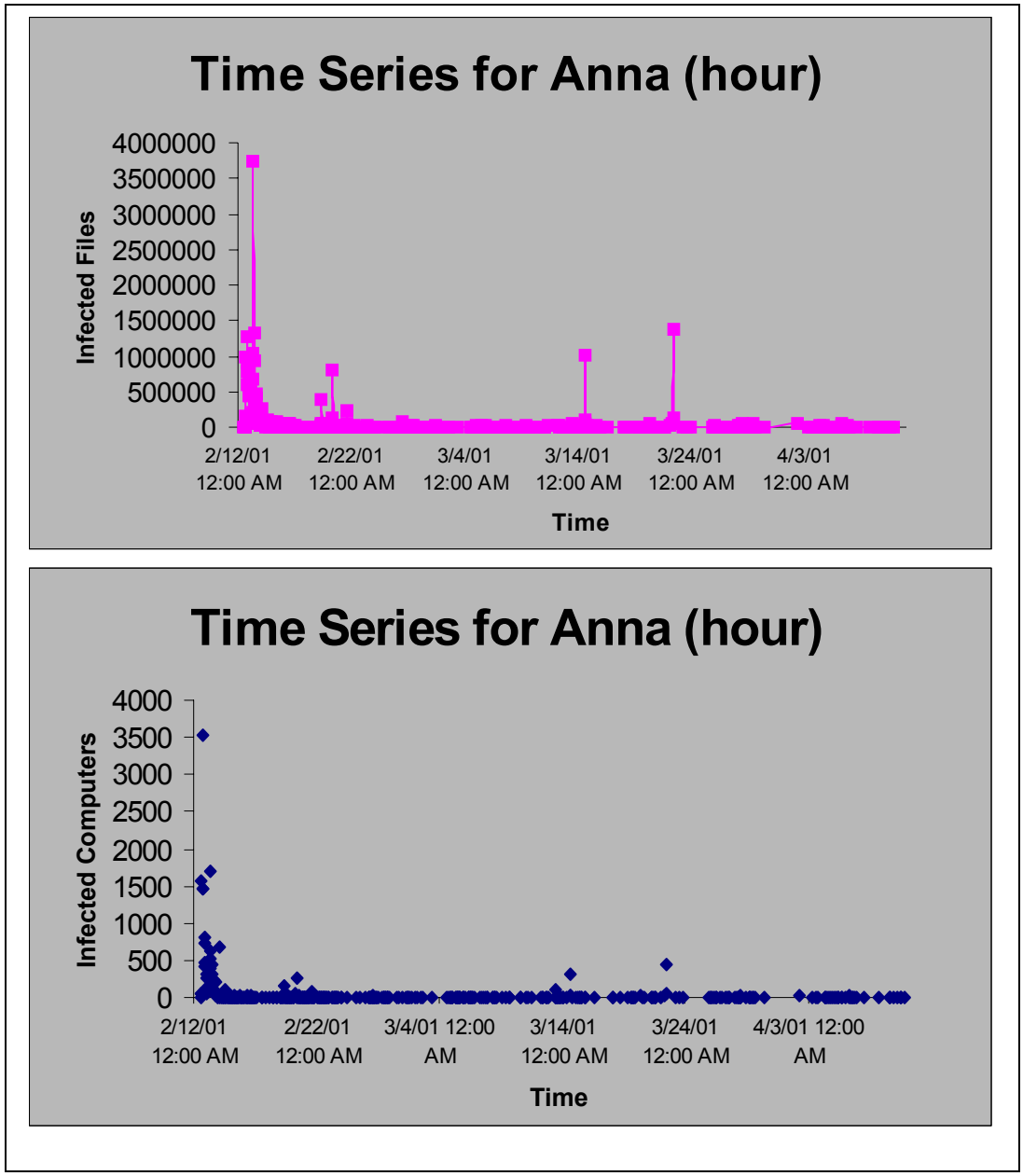
C.5.5 Doubling time

The estimated doubling time for Anna is 2.6 hours. One million infections would be achieved between 8 and 9 generations (20.9 – 23.4 hours).

C.5.6 Anti-virus data

The data in Figure C-12 capture the initial epidemic rise and endemic tail for Anna. The graph shows that Anna peaked less than 24 hours after release and that the peak was very brief.

Figure C-12. Number of infections per hour for Anna reported by Anti-Virus Company 2 (expressed as number of infected files (top) and number of infected machines (bottom))



C.6 Kak

C.6.1 Technical description

The following technical description was published by F-Secure Corporation at http://www.europe.f-secure.com/v-descs/kak.htm.

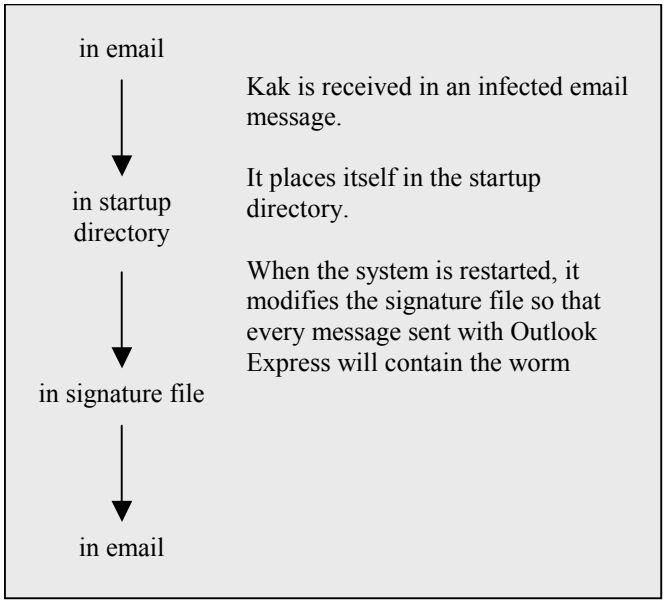
NAME: Kak ALIAS: Wscript.KakWorm, KakWorm	ANNOTATIONS
Kak is a worm that - like BubbleBoy - embeds itself without any attachment to every e-mail sent from the infected system. For further information about BubbleBoy, see the description: http://www.F-Secure.com/v-descs/bubb-boy.shtml	
Kak is written in JavaScript and it works on both English and French versions of Windows 95/98 if Outlook Express 5.0 is installed. It does not work in a typical Windows NT installation.	
The worm uses a known security vulnerability that is in Outlook Express. Once the user receives an infected e-mail message and opens or views the message in the preview pane, the worm creates a file "kak.hta" to the Windows Startup directory.	Email must be read or previewed in Outlook Express to infect. Execution of an email attachment is not required. Reboot is required.
Next time the system is restarted, the worm activates. It replaces "c:\autoexec.bat" with a batch file that deletes the worm from the Startup directory. The original "autoexec.bat" is copied to "C:\AE.KAK".	
Also, It modifies the message signature settings of Outlook Express 5.0 by replacing the current signature with an infected file, "C:\Windows\kak.htm".	The worm is attached to all legitimate emails sent after infection.
Therefore every message sent with Outlook Express will contain the worm after this has been done.	Runs continuously until disinfection.

Then it modifies the Windows registry in such a way that it will be executed in every system startup. The key it adds to the registry is: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\cg0u The .hta file that the virus creates and executes in the future is saved to Windows System directory. On the first day of each month, if the number of hours is more than 17 (i.e. 6pm or later), the worm will show an alert box with the following text: Kagou-Anit-Kro$oft say not today! Then the worm shuts down Windows.	Payload is not included in BRR calculation.

C.6.2 Life cycle

The conceptualized life cycle for Kak is shown in Figure C-13

Figure C-13. Life cycle of Kak

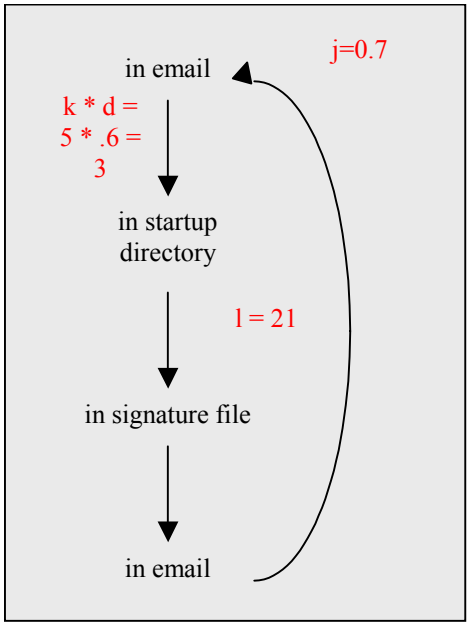


C.6.3 Basic reproduction rate (BRR)

Because Kak spreads automatically and attaches itself through legitimate emails, it may evade detection for relatively long periods of time. This increases the number of cases that one infected case can produce. We assume that the email will send one email every eight hours for one week before detection on average. Email must be opened or previewed in OutLook (probability = 0.6). The parameters used to calculate the BRR for Kak are shown in Figure C-14. The estimated BRR for Kak is 0.7 *5 *.6 * 21 = 44.1.

Parameter descriptions:
d=P(recipient does not delete file on arrival)
j=P(use Outlook and Windows)
k=mean number of recipients per legitimate email
l=mean number of legitimate email messages

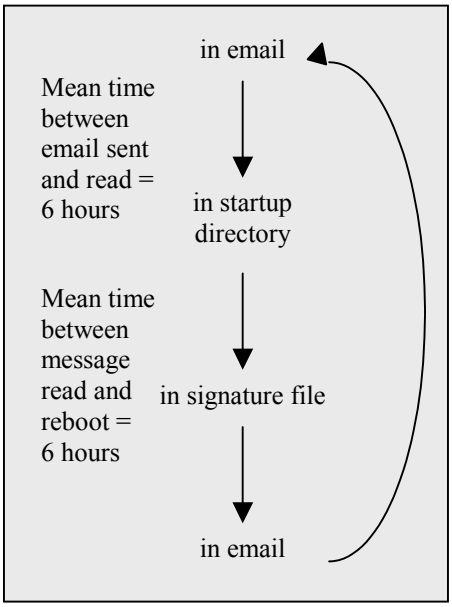
Figure C-14. Basic reproduction rate parameters for Kak



C.6.4 Generation time

There are two delays in the life cycle that result in an estimated generation time of 12 hours. These are shown in Figure C-15.

Figure C-15 Generation time diagram for Kak



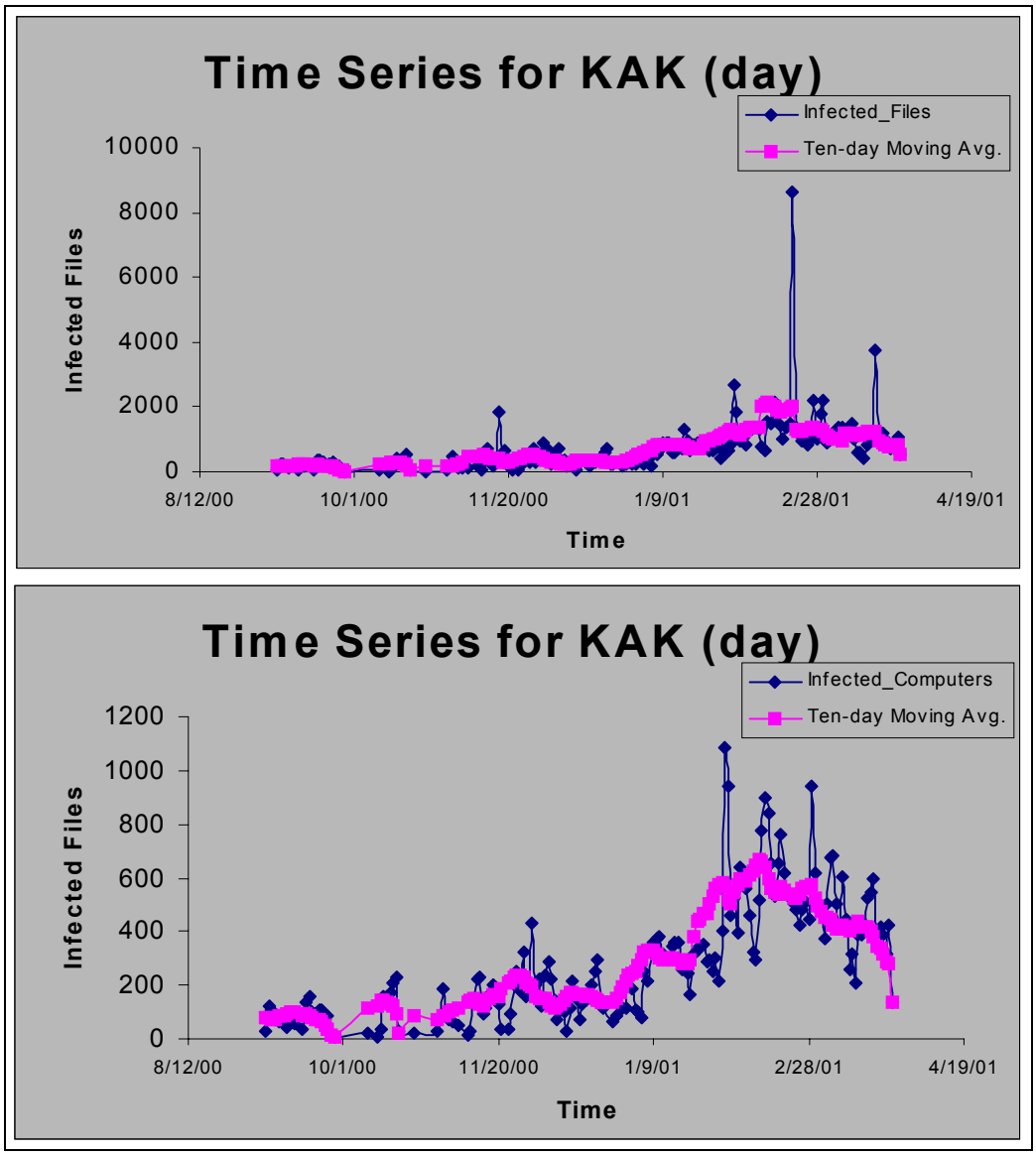
C.6.5 Doubling time

The estimated doubling time for Kak is 2.2 hours. This estimate disagrees with real world data which shows very slow growth. This may be due to several reasons. Kak was released in October, 1999[16]. Public awareness was heightened by a media report of a large scale release in May, 2000, when a mass mailing by online computer retailer Shoppingplanet.com sent the infection to 50,000 customers. Effective controls may have depressed the epidemic potential of this worm. Kak may be an example of effective defense against a very precocious worm.

C.6.6 Anti-virus data

[16] http://vil.nai.com/vil/virusSummary.asl?virus_k10509

Figure C-16. Number of infections per day for Kak reported by Anti-Virus Company 2 (expressed as number of infected files (top) and number of infected machines (bottom))



The real world data from Anti-Virus Company 2 given in Figure C-16 for Kak reflect its persistent endemic occurrence in the population. These data show it continues to spread a year after initial release, but slowly. Prominent features such as the elevated report rate in February 2001, are probably not indicative of the worm's intrinsic behavior.

C.7 MTX

C.7.1 Technical description

The following technical description was published by F-Secure Corporation at http://www.europe.f-secure.com/v-descs/mtx.htm.

NAME: MTX ALIAS: IWorm_MTX, I-Worm.MTX, Matrix ALIAS: Apology, W32/Apology	ANNOTATIONS
The MTX worm has three components - worm, virus and backdoor. It spreads under Win32 systems - the virus component infects Win32 executable files, attempts to send e-mail messages with infected attachments and installs the backdoor component to download and spawn "plugins" on an affected system.	Two transmission paths: worm and virus. Backdoor is not included in BRR calculation.
The virus has an unusual structure. It consists of three different components that are run as standalone programs (Virus, email Worm and Backdoor). The virus is the main component, it keeps the worm and the backdoor programs in its code in compressed form. While infecting the system, it extracts and spawns them:	Installation of virus is a prerequisite to infection by worm and backdoor.
The MTX worm-virus structure looks like this: ---------- I The virus I --> installs Worm and Backdoor to the system, I installation I then finds and infects Win32 executable files I and infection I I routines I ---------- I Worm code I --> is extracted to file and run as stand-alone program I (compressed) I	

---------- I Backdoor code I --> is extracted to file and run as stand-alone program I (compressed) I ---------- The worm code does not contain all the necessary routines to infect the system where the infected e-mail (see below) is sent as an attachment. The worm file is infected by the virus as an ordinary file and then sent. The reason to use such a way is not clear. Probably the components were written by different people. The Virus component contains the following text strings: SABIÁ.b ViRuS Software provide by [MATRiX] VX TeAm: Ultras, Mort, Nbk, LOrd DArk, Del_Armg0, Anaktos : All VX guy in #virus and Vecna for help us Visit us at: http://www.coderz.net/matrix The worm component contains the following text strings: Software provide by [MATRiX] VX team: Ultras, Mort, Nbk, LOrd DArk, Del_Armg0, Anaktos Greetz: All VX guy on #virus channel and Vecna Visit us: www.coderz.net/matrix The Backdoor contains the following text strings: Software provide by [MATRiX] team: Ultras, Mort, Nbk, LOrd DArk, Del_Armg0, Anaktos Greetz:Vecna 4 source codes and ideas Virus Component	

The virus component uses EPO (Entry Point Obscuring) technology while infecting a file. This means that the virus does not affect the file at its entry code, but places "Jump-to-Virus" instruction somewhere in the middle of the file code section to make the detection and disinfection procedures more complex. As a result the virus is activated only if the corresponding affected program's branch receives control.

The virus is also encrypted, so first of all it decrypts itself when its code gets control. The virus then looks for necessary Win32 API functions by scanning Win32 Kernel. The virus tries Win9x, WinNT and Win2000 addresses to do this.

The virus then looks for anti-virus programs active in the system and exits if any of them is detected. The list of anti-virus programs the virus looks for is as follows:

AntiViral Toolkit Pro
AVP Monitor
Vsstat
Webscanx
Avconsol
McAfee VirusScan
Vshwin32
Central do McAfee VirusScan

Then the virus installs its components to the system. They are decompressed installed to the Windows directory and then spawned. Three files created in there with the hidden attribute set. Their names are:

IE_PACK.EXE - pure Worm code
WIN32.DLL - Worm code infected by the virus
MTX_.EXE - Backdoor code

The virus then infects Win32 executable

Virus will not install if anti-virus software is present.

Virus installs to Windows directory.

PE EXE files in current, temporary, and Windows directories, and then exits. Worm Component The worm component uses technology that was first introduced by Happy99/Ska Internet worm to send infected messages. The worm affects WSOCK32.DLL file in the Windows system directory by appending a component of its code to the end of the file and hooking the "send" WSOCK32.DLL routine. As a result, the worm monitors all data that is send from an affected computer to the Internet. Usually WSOCK32.DLL file is in use at the moment the worm starts and it is locked by Windows. To avoid that, the worm uses the standard way: it creates a copy of the original WSOCK32.DLL with the name WSOCK32.MTX, infects that copy and then writes "replace original file with infected" instructions to the WININIT.INI file: NUL=C:\WINDOWS\SYSTEM\WSOCK32.DLL C:\WINDOWS\SYSTEM\WSOCK32.DLL=D:\WINDOWS\SYSTEM\WSOCK32.MTX where "C:\WINDOWS\SYSTEM" is the name of the Windows system directory and may differ depending on the name of the directory where Windows is installed. The infected WSOCK32 replaces the original one during the next reboot, and the worm gets access to data that is sent from the infected machine. The worm pays attention to the Internet sites (Web, ftp) that are visited, as well as to e-mail messages that are sent from the computer.	Reboot required.

The most visible behaviour of the virus is that it stops visiting several Internet sites and disables sending messages to the same domains (they are anti-virus domain names). The virus detects them by four-letter combinations: nii. nai. avp. f-se mapl pand soph ndmi afee yenn lywa tbav yman Furthermore, the worm does not allow user to send e-mail messages to the following domains: wildlist.o* il.esafe.c* perfectsup* complex.is* HiServ.com* hiserv.com* metro.ch* beyond.com* mcafee.com* pandasoftw* earthlink.* inexar.com* comkom.co.* meditrade.* mabex.com * cellco.com* symantec.c* successful* inforamp.n* newell.com* singnet.co*	Two emails sent to all recipients of legitimate emails sent from infected computer.

bmcd.com.a*
bca.com.nz*
trendmicro*
sophos.com*
maple.com.*
netsales.n*
f-secure.c*

The worm also intercepts e-mail messages that are sent and attempts to send a duplicate message with the infected attachment to the same address (the same as "Happy99/Ska" worm does). As a result, victim address should receive two messages: first is the original message written by the sender, second is a message with empty subject and text and attached file that has one of the names that are selected by worm depending on current date:

README.TXT.pif
I_wanna_see_YOU.TXT.pif
MATRiX_Screen_Saver.SCR

LOVE_LETTER_FOR_YOU.TXT.pif

NEW_playboy_Screen_saver.SCR
BILL_GATES_PIECE.JPG.pif
TIAZINHA.JPG.pif
FEITICEIRA_NUA.JPG.pif
Geocities_Free_sites.TXT.pif
NEW_NAPSTER_site.TXT.pif
METALLICA_SONG.MP3.pif
ANTI_CIH.EXE

INTERNET_SECURITY_FORUM.DOC.pif
ALANIS_Screen_Saver.SCR

READER_DIGEST_LETTER.TXT.pif
WIN_$100_NOW.DOC.pif

IS_LINUX_GOOD_ENOUGH!.TXT.pif
QI_TEST.EXE
AVP_Updates.EXE

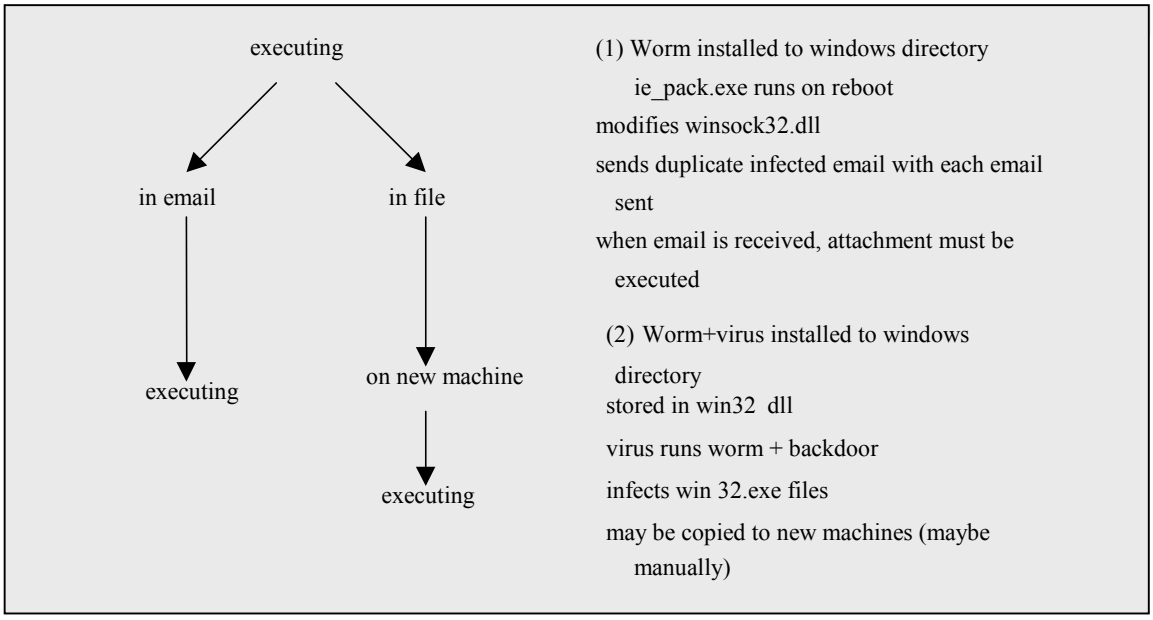
SEICHO-NO-IE.EXE YOU_are_FAT!.TXT.pif FREE_xxx_sites.TXT.pif I_am_sorry.DOC.pif Me_nude.AVI.pif Sorry_about_yesterday.DOC.pif Protect_your_credit.HTML.pif JIMI_HMNDRIX.MP3.pif HANSON.SCR XXXX_WITH_DOGS.SCR MATRiX_2_is_OUT.SCR zipped_files.EXE BLINK_182.MP3.pif The worm sends out the WIN32.DLL file that was dropped by the virus component during MTX's first installation to the infected system. Note: the worm does not drop WIN32.DLL file, but uses that file to attach it to messages that are sent. So the "pure worm" is not able to spread more than once: when run on victim machine it will infect WSOCK32.DLL, but will not able to send its copies further. To "fix that problem" the worm sends its infected copy (WIN32.DLL is the worm component infected by the virus component, see above). Fortunately, the known worm modification has a bug in its spreading routine and the e-mail server fails to receive affected messages from the infected machine. So, the known worm version cannot be widely spread. Backdoor Component Being run, the backdoor component creates a new key in system registry that indicates that the machine is already infected: HKLM\Software\[MATRIX]	Backdoor not included in BRR calculations.

If this key exists, the Backdoor skips the installation procedure. Otherwise it registers itself in auto-run section: HKLM\Software\Microsoft\Windows\CurrentVersion\Run SystemBackup=%WinDir%\MTX_.EXE where %WinDir% is Windows directory. The backdoor then stays active in Windows as a hidden application (service) and runs a routine that connects to some Internet server, gets files from there and spawns them to the system. So the Backdoor can infect the system with other viruses or install trojan programs or more functional backdoors. The backdoor in the known virus version has a bug that causes a standard error message when the backdoor tries to access the Internet site.	

C.7.2 Life cycle

The conceptualized life cycle for MTX is shown in Figure C-17.

Figure C-17. Life cycle of MTX



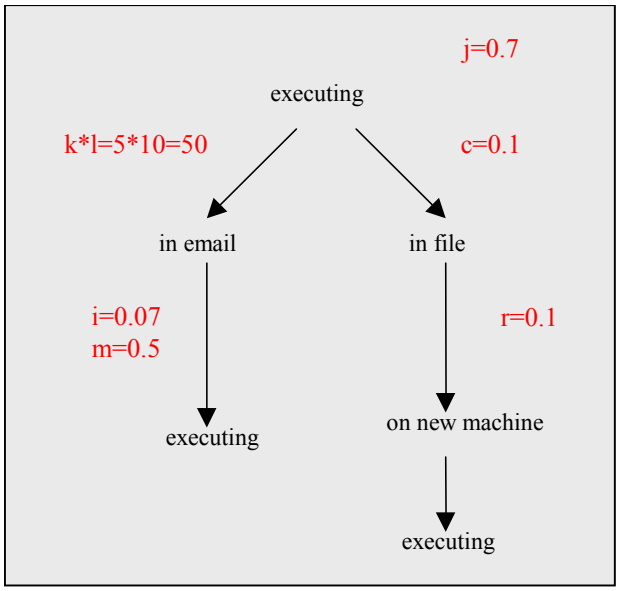
C.7.3 Basic reproduction rate (BRR)

In calculating an estimate of BRR for MTX, we make the following modifications to the default parameters:

- P(attachment will be opened) is reduced to 0.07, reflecting the increased likelihood of detection that results when the worm sends two emails, one infected and one not, to every legitimate recipient of email.
- P(anti-virus software is present) has been added. The virus will not install the worm in this case.

These parameters are shown in Figure C-18.

Figure C-18. Basic reproduction rate parameters for MTX



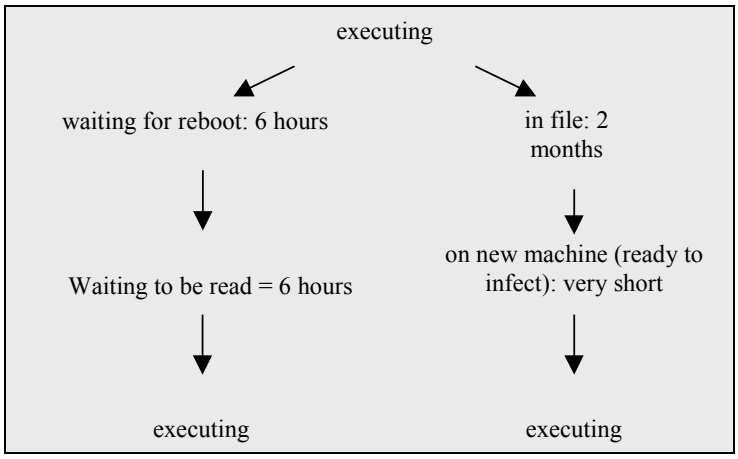
Parameter description:
c=P(file being transferred to another machine)
i=P(attachment will be opened)
j=P(use Outlook and Windows)
k=mean number of recipients per legitimate email
l=mean number of legitimate email messages
m=P(virus does not exit due to finding anti-virus software)
r=P(sent file is infected)

The estimated BRR for MTX = BRR (email) + BRR (in file)
= k*l*I*m + cr
= 1.225 + .007
= 1.232

C.7.4 Generation time

Estimated generation times for the two transmission paths are 12 hours for email and 2 months for in file (Figure C-19).

Figure C-19. Generation time diagram for MTX



C.7.5 Doubling time

The estimated doubling time for MTX is 41 hours. This is approximately half the doubling time of seven to eight days shown in the data obtained from Anti-Virus Company 1. This doubling time is based on a BRR of 1.24, which is very close to the BRR or 1.0 indicating control.

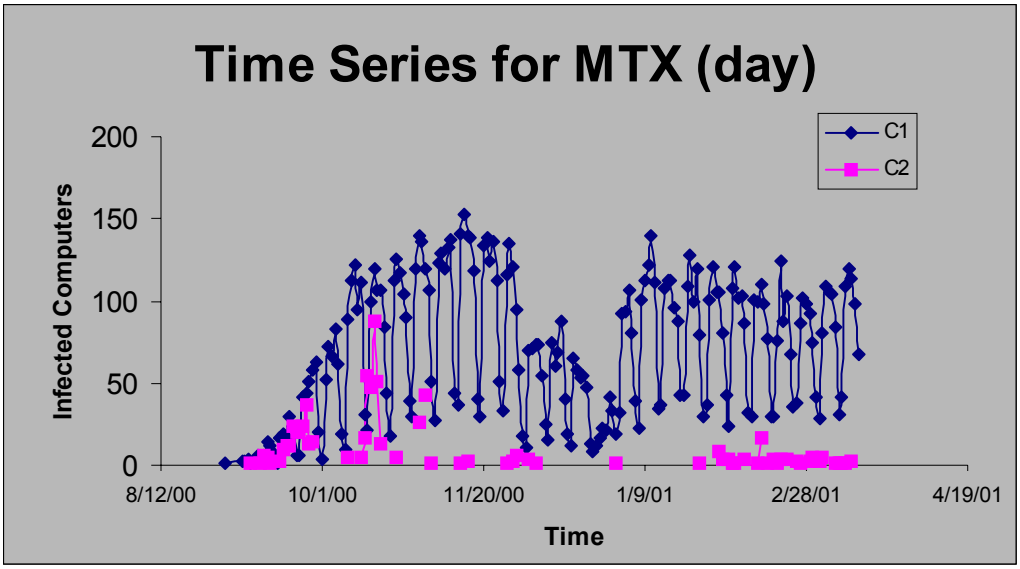
C.7.6 Anti-virus data

In the graph shown in Figure C-20, MTX was first discovered on 17 August 2000 (Symantec).
The time period shown includes the initial growth curve.

This graph illustrates a number of interesting features:

- A characteristic sigmoid curve, characteristic of logistic growth. The initial portion of the curve is approximately exponential and shows a doubling time of 7 or 8 days.
- After the initial increase, Company 1's curve remains high, although it is very noisy and probably decreasing slowly.
- Company 1's data show a very clear weekly cycle, which usually peaks on Tuesdays.
- A seasonal decrease around Christmas is also clear.
- Company 2's data track Company 1's data through the exponential portion of the curve. The remainder of the curve is relatively uninformative, and is not well correlated with Company 1's data.

Figure C-20. Number of infections per day for MTX reported by Anti-Virus Companies 1 and 2



C.8 Ethan

C.8.1 Technical description

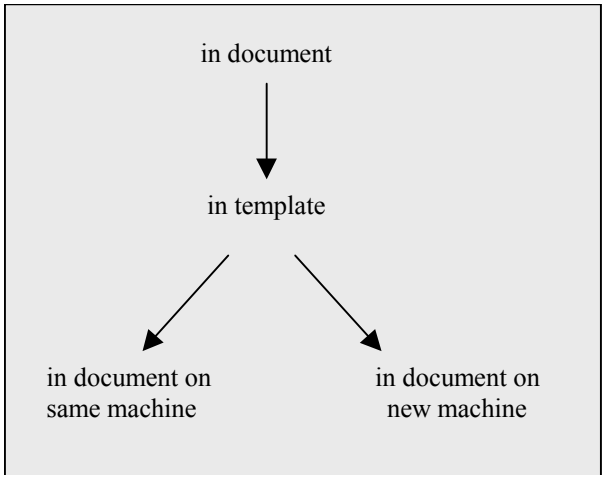
The following technical description was published by McAfee at http://www.vil.nai.com/vil/virusSummary.asp?virus_k=10107.

W97M/Ethan.A is a Word97 Macro Virus. It is a fast moving infector and reported to numerous AVERT Labs around the globe. Infection takes place when an infected Word document is closed, allowing the virus to propagate itself to normal.dot template. W97M/Ethan.A is a parasitic class module infector, which consists of one macro, and is approximately 50 lines of code in length. It infects documents and templates using an algorithm to input data, from a source file c:\ethan.___ to the host document. This source file is exported VBA code of the virus. Viruses using class module infection method transfer the virus code to the "ThisDocument" container. This virus prepends its code and infects all documents accessed. There is a 3-in-10 chance that this virus will modify the document properties of infected files: Title = "Ethan Frome" Author = "EW/LN/CB" Ethan has one additional characteristic in that if it detects the "class.sys" file (created by the W97M/Class virus infection) on the machine it will delete it.	ANNOTATIONS The macrovirus infects a MS Word template. This probability is not included in the BRR calculation because it does not pertain to replication.

C.8.2 Life cycle

The conceptualized life cycle for Ethan is shown in Figure C-21. We assume that the virus will be transferred to a new host through an infected document attached to email.

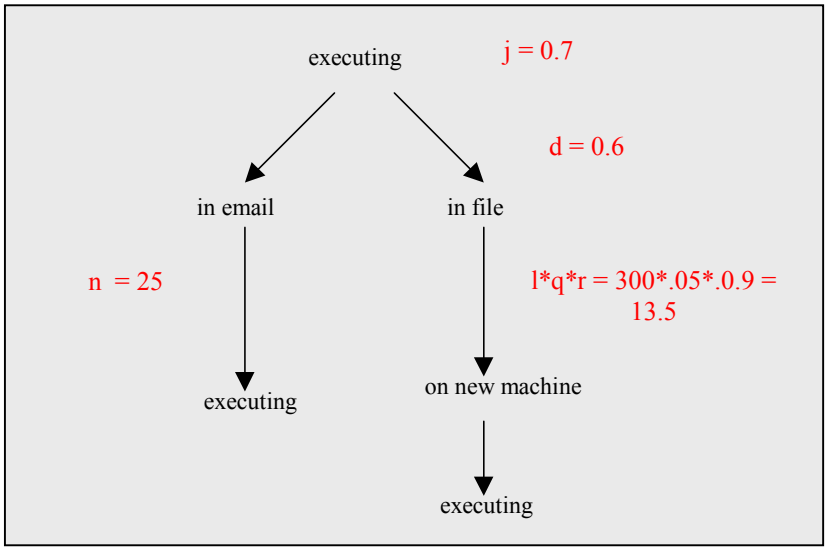
Figure C-21. Life cycle of Ethan



C.8.3 Basic reproduction rate (BRR)

Like Kak, the infectious spread of this macrovirus continues until it is detected. We assume that the virus will escape detection for two months. The estimated BRRs for the two transmission paths, BRR (same machine) and BRR (new machine) are 17.5 and 5.7 respectively, resulting in a BRR (total) of 23.2. These parameters are illustrated in Figure C-22.

Figure C-22. Basic reproduction rate parameters for Ethan

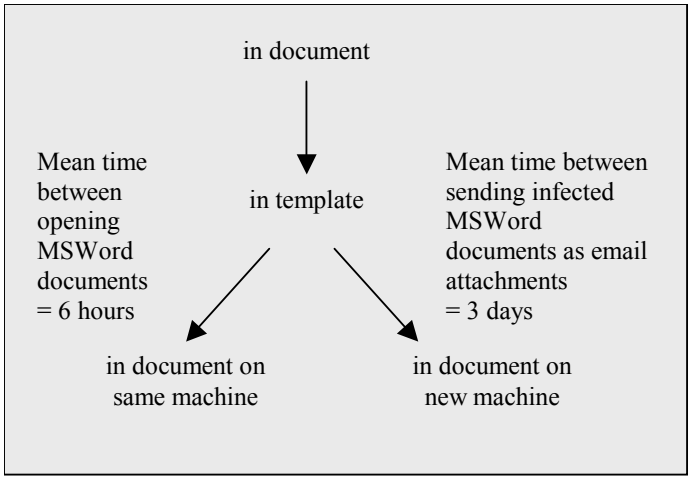


Parameter description:
d=P(recipient does not delete file on arrival)
j=P(use Outlook and Windows)
l=mean number of legitimate email messages
n=number of uninfected documents on a host
q=P(email contains MS Word attachment)
r=P(sent file is infected)

C.8.4 Generation time

Estimated generation time for Ethan is shown in Figure C-23. Generation time for the same-host branch is 6 hours; for the new-host branch, 72 hours.

Figure C-23. Generation time diagram for Ethan



C.8.5 Doubling time

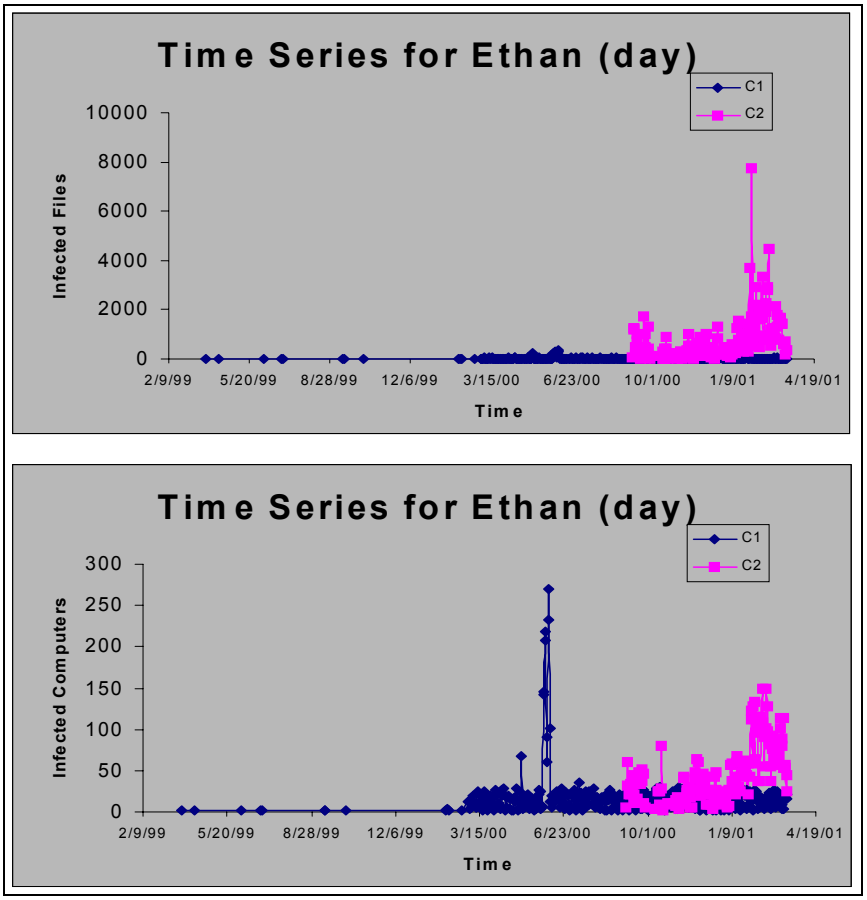
The estimated generation times for the two transmission paths are:
Doubling time (same host) = 1.5 hours;
Doubling time (new host) = 28.8 hours.

These doubling times indicate that while infection on any one machine occurs relatively rapidly, infection among machines proceeds at a slower pace.

C.8.6 Anti-virus data

Data reported regarding the spread of Ethan are shown in Figure C-24. Several regimes are clearly visible, but the transitions appear to be unrelated to the macrovirus's

Figure C-24. Number of infections per day for Ethan reported by Anti-Virus Company 2 (expressed as number of infected files (top) and number of infected machines (bottom))



intrinsic behavior. The sudden increases in incidence may be due to rare events, such as mass mailings of infected files, but we have no evidence to substantiate this hypotheses.