BIOLOGICAL APPROACH TO SYSTEM INFORMATION SECURITY (BASIS)

A New Paradigm in Autonomic Information Assurance

FINAL REPORT

CONTRACT #30602-01-0509

PERIOD OF PERFORMANCE: February 28 – September 31, 2002

Prepared by
Victor A. Skormin, Ph.D.
Principal Investigator, Professor of Electrical Engineering,
Watson School of Engineering, Binghamton University, Binghamton NY
Telephone: (607)777-4013, -4856 (secretary), E-mail: vskormin@binghamton.edu

Contractual POC: **MaryBeth Curtin**
Grant & Contract Administrator, Office of Research & Sponsored Programs
Binghamton University
Telephone: (607)777-6136, E-mail: mcurtin@binghamton.edu

PREPARED FOR:

AFRL Program Manager:
Joseph Giordano
Air Force Research Laboratory/IFGB
525 Brooks Road, Rome NY 13441-4505,
Phone: 315-330-4199, Fax: 315-330-4390

Grants Administration Office:
ONRRO Boston
495 Summer St., Rm 627
Boston, MA 02210-2109

AFRL Grants Administrator:
Gail E. Marsh
Air Force Research Laboratory/IFKE
26 Electronic Parkway
Rome NY 13441-4514
Phone: 315-330-1940, Fax: 315-330-8033

The Research Team

Victor Skormin, Ph.D., Principal Investigator, Professor, Electrical and Computer Engineering, Binghamton University
Douglas Summerville, Ph.D., co-PI, Assistant Professor, Electrical and Computer Engineering, Binghamton University
Dennis McGee, Ph.D., Investigator, Associate Professor, Biology Department, Binghamton University
James Moronski, Investigator, Doctoral Student, Electrical and Computer Engineering, Binghamton University

Acknowledgments

The research team of the BASIS project is grateful to:

Leonard Popyack, Ph.D., of the Air Force Research Laboratory at Rome NY for fruitful discussions leading to the formulation of the BASIS concept

Joseph Giordano, **John Graniero** and **Donald J. Nicholson,** Ph.D., of the Air Force Research Laboratory at Rome NY for their on-going support of the BASIS project

Alexander Tarakanov, Sc.D., and **Svetlana Sokolova**, Sc.D., of the Institute of Information and Control Technologies, St. Petersburg Section of the Russian Academy of Sciences for their contribution to this research and the discussion of the results

Alexander Polevschikov, Sc.D., of the Institute of Experimental Medicine, St. Petersburg Section of the Russian Academy of Sciences for his expert advice in the area if modern immunology

Vladimir Gorodetski, Sc.D., of the Institute of Information and Control Technologies, St. Petersburg Section of Russian Academy of Sciences for fruitful discussions and advice on multi-agent methodology and cluster analysis

Dipankar Dasgupta, Ph.D., Associate Professor of Computer Engineering at the University of Memphis, TN for the discussion of the results and friendly advice on various aspects of computer immunology

Ann Miller, Ph.D., Distinguished Professor of Computer Engineering, University of Missouri-Rolla for friendly discussions of the research plans

Jiri Hodik, Visiting Researcher, Ph.D. Student at the Technical University of Prague for his efforts in applying the system concepts to describing the immune response

Contents

I. Introduction

With the increase of size, interconnectivity, and number of points of access, computer networks have become vulnerable to various forms of information attacks, especially to new, sophisticated ones. It should be pointed out that biological organisms are also complex and interconnected systems that have many points of access; these systems are vulnerable to sabotage by alien microorganisms that, based on the attack mechanisms described in modern immunology, could be viewed as information attacks. During evolution, biological organisms have developed very successful immune systems for detecting, identifying, and destroying most alien intruders. This research is aimed at the establishing a connection between the basic principles that govern the immune system and potential uses of these principles in the implementation of information security systems (ISS) for computer networks. This is the goal of the proposed Biological Approach to System Information Security (BASIS) that is presented in this report.

The following table features some basic similarities between biological and computer defense systems. However, a mutually beneficial cross-pollination between information security and immunology have been effectively prevented by historically different methodologies:

Table 1. Similarities Between Biological and Computer Defense Systems

Biological Systems	Computer Networks
High complexity, high connectivity, extensive interaction between components, numerous entry points.	High complexity, high connectivity, extensive interaction between components, numerous entry points.
Vulnerability to intentionally or unintentionally introduced alien microorganisms that can quickly contaminate the system resulting in its performance degradation and collapse.	Vulnerability to malicious codes (including computer viruses) that being introduced in the system result in unauthorized access to information and services and/or denial of service.
Alien microorganisms as well as cells of a biological system are composed of the same building blocks - basic amino acids.	Malicious codes as well as the operational software of a computer network are composed of the same building blocks - basic macro commands.
The difference between alien microorganisms and the healthy cells of a biological system is in the (gene) sequencing of their building blocks.	The difference between malicious codes and the operational software of a computer network is in the sequencing of their building blocks.
Biological immune systems *are capable of* detecting, recognizing and neutralizing most alien microorganisms in a biological system.	Information security systems *should be capable of* detecting, recognizing and neutralizing most attacks on a computer network.

- biological immunology is a discipline dominated by chemistry, with unsophisticated mathematical apparatus and primarily qualitative approach, addressing intimidating complexity of the living tissue (cannot contact the designer)

- information security is dominated by mathematics and quantitative approach, it deals with relatively simple material base (always can contact the designer and manufacturer)

Computer immunology [1], [2], [3] is a very new direction in computer information security aimed at the exploration and possible utilization of various aspects of immunology for the development of novel information security systems. The BASIS project is a multi-disciplinary effort facilitating an exchange of methodologies, concepts and philosophies between two disciplines. The approach proposed in this research, reflects the similarities between the computer network security problem and the task of protecting a biological system from invading microorganisms. Its ultimate goal is to synthesize an ISS of a computer network that follows the basic principles of operation of the biological immune system.

In order to be dependable, existing ISS must be able to prevent particular kinds of threats and suppress most specific types of attacks. This requires an enormous amount of data processing that adversely affects the overall network performance and throughput. With the complexity of modern information security systems, an ISS must comprise a large number of independent, largely autonomous, network-based, specialized software agents operating in a coordinated and cooperative fashion designated to prevent particular kinds of threats and suppressing specific types of attacks. This is the only realistic approach providing the required level of general security of information according to a global criterion without burdening the network resources. It could be achieved only by adopting the most advanced principles of interaction between particular agents. The utilization of biological defensive mechanisms developed by million-year evolution has a great potential for the assurance of information security in large-scale computer networks. Immune defense mechanisms have distributed cells (agents) of various types that attack anything suspected to be alien. Cells interact by sharing information about the type and location of an intruder, utilize the feedback principle for engaging only “as many cells as necessary,” and are capable of learning about intruders that results in immunity to repeated attacks. Research in immunology has established various mechanisms of individual behavior of cells resulting in their ability to detect, identify, pursue and destroy an alien entity; to accumulate knowledge on attackers, to adopt behavior to a new situation; and to determine the proper response. These mechanisms, developed by evolution, are highly efficient and successful. In addition to individual cell operation, immunology presents numerous examples of collective, almost intelligent, “unselfish” behavior of various types of defensive cells. This collective cell behavior allows the achievement of high efficiency and minimum response time of the immune system, as well as maximum utilization of its limited resources. The major difference in the targets between the immune system and an ISS is: the immune system treats as an “enemy” *any foreign entity* within the organism, while an ISS must recognize and treat as an “enemy” *any illegitimate entry* or *software*.

A biological immune system of an advanced organism already has been considered as a good and clear example of a modern agent-based ISS [4], [5]. The *immune system* consists of distributed white blood cells, which attack anything that they consider alien. By having as many cells as necessary, the animal body is able to defend itself in a very efficient way. If the animal body is infected in one area, then cells move to that area and defend it. Modern multi-agent system technology presents a valuable approach for the development of an ISS that is expected to have very promising advantages when implemented in a distributed large-scale multi-purpose information system. A consideration of an agent-based model of an ISS, consistent with the BASIS concept, is given in [6].

The BASIS project, however, does not pursue the development of an artificial immune system that at the present level of technology is not feasible [7]. The research is aimed at the selection of particular information security problems, common in computer networks. Then, based on the expertise in the operation of the immune system, similar situations common in the immune system are to be identified, and the mechanisms of the immune response will be investigated. The description of the appropriate immune response is obtained, first on the qualitative level, and then the mathematical models will be obtained. Implemented in software, these models could be used as simulation tesbeds suitable for the analysis of the immune response and establishing the conditions leading to particular outcomes. Finally, an attempt will be made to adopt these models for the design of defense mechanisms intended for a computer network.

Our ISS approach can be viewed as the development of a set of semi-autonomous distributed agents capable of detecting, recognizing, pursuing, and learning about the attackers. Necessary functions of the ISS agents addressing various tasks within the ISS are to be established. While the implementation of particular ISS agents may be computationally intensive, a feasible ISS system requires a high degree of interaction, coordination, and cooperation between agents. It is believed that these, almost intelligent, interactions constitute the centerpiece of a biological defensive mechanism and assure its high efficiency. Until recently, quantitative description of the cooperative behavior of semi-autonomous agents presented a problem that could not be solved within a general framework. Such a framework has been provided by the multi-agent system approach that can be utilized as a mathematical apparatus to facilitate the formalization of the complex interaction between particular agents in the fashion that is observed in biological immune systems. This facilitates the development of the rules and procedures of the interaction between ISS agents thus leading to the development of a feasible ISS reflecting the operation of an immune system. The feasibility of such an ISS could be assured only by the implementation of the rules of agents' collective behavior observed in a biological immune system that could be formalized using the multi-agent system theory.

A number of mathematical models of individual behavior of defensive cells utilizing methods of statistics, discrete mathematics, and numerical simulation, have been successfully implemented. However, only [8] provides a comprehensive, rigorous mathematical basis for the quantitative description of the operation of immune cells. The attempts to develop an artificial immune system that could be applied for such a practical problem as information security assurance are much less successful, primarily because of the necessity to describe mathematically cooperative cell behavior. However, modern multi-agent system technology presents the most plausible approach for solving this problem in the framework of the development of an ISS. The resultant ISS would operate as a synthetic immune system and is expected to have very promising advantages when implemented in a distributed large-scale multi-purpose computer information network.

We propose the following steps to achieve ISS by means of biologically inspired schemes:

Analysis and Qualitative Description of Immune Systems. An analysis of the recent biological research and development of a comprehensive qualitative description of the operation of a biological immune system are needed to capture the behavior immune system as it may relate to ISS-related problems.

Algorithmic Description Immune Cells. The next step is to develop a mathematical/ algorithmic description of the individual behavior of immune cells utilizing already established methods and models, and their cooperative operation using the multi-agent system theory.

Software Implementation of Models. A software implementation of the established mathematical models, rules and algorithms resulting in an ISS operating as an artificial immune system is necessary to test and check these models.

Simulation Environment. A simulation environment suitable for the representation of a computer network with a resident ISS and various forms of threats and attacks needs be developed to prove the model.

System Analysis. Simulations need to be analyzed to fine-tuning of the resultant ISS, This analysis should include assessment of its impact on the network vulnerability.

Particular tasks of the BASIS project addressed during the performance period were:

1. Formulation of the specific features of the *denial of service* and malicious code-type attacks on the computer network. Definition of their biological equivalents, such as bacterial infection, viral infection, food poisoning, etc.

2. Analysis of the general biological defense strategies and defensive mechanisms with the emphasis on dynamics of major developments, particular immune responses/agents involved, communication between agents of the same types, communication between agents of different types, communication channels: forms and formats, cloning and repositioning of agents, agent cooperation, task distribution, resource allocation.

3. Application of the multi-agent framework for the description of the system architecture, functions of particular agents, coordination, cooperation, task distribution, and resource allocation tasks, addressing such system/agent properties as distributability, multi-barrier mechanism, self-rule (autonomous) cells, adaptability and memory. The particular agent lists includes access control agents, audit and intrusion detection agents, anti-intrusion agents, diagnostic and information recovery agents, cryptographic and steganography agents, authentication agents, meta-agents

4. Development of an experimental heterogeneous computer network – testbed suitable for the simulation analysis of attacks and ISS, and the auxiliary software

While the BASIS project was intended as a three-year effort, this report reflects only the first 18 months effort.

Section II of this Report provides the description of the multidisciplinary, international project team including areas of specialization, qualifications and affiliations of its particular members. This team has been established at several international conferences in the area of computer information security centered around research activities sponsored by the AFRL at Rome NY and the European Office of Aerospace Research and Development of the USAF operating in London, UK.

Section III of the Report describes major types of the attacks on a computer network, primarily Denial of Service (DoS) attacks and attacks by the deployment of malicious codes, viruses, worms and Trojan Horses. This description provides the justification for the search of novel solutions of the computer network security problem. It also is important for making an argument on the similarity between biological and computer attacks and attackers.

Section IV of the Report provides a comprehensive description of the operation of the immune system with the emphasis on the particular attackers, specific agents of the immune system, their individual and collective behavior, cooperation, task distribution, information exchange, consistent with the most recent findings of immunology.

Section V presents further discussion of the similarity between computer and biological attacks, attackers, and defense mechanisms. Provided "anatomies" of typical attacks show the ways to establish immune-like information security systems. An attempt is made to describe the dynamics of the immune response leading to a lethality, full recovery or chronic disease of a biological organism and to utilize the same equations for the formulation of control laws that could be recommended for computer networks.

Section VI provides mathematically rigorous approach, utilizing singular-value decomposition as the underlying computational mechanism, for the solution of a typical computer security problem, the intrusion detection. Methods of immunocomputing formulated in [8] are employed for pattern recognition, supervised and unsupervised learning, and the analysis of the specifics of the intrusion with the potential for the identification of the intruder.

Section VII describes the authors' research in the area of attack detection by monitoring and mining of the data traffic in computer networks. This research has resulted in a novel data mining approach utilizing statistical clustering, genetic optimization, and Bayesian estimation. Unlike existing results in this area, it provides explicit attack probabilities and could be recommended as a decision support tool for a network operator.

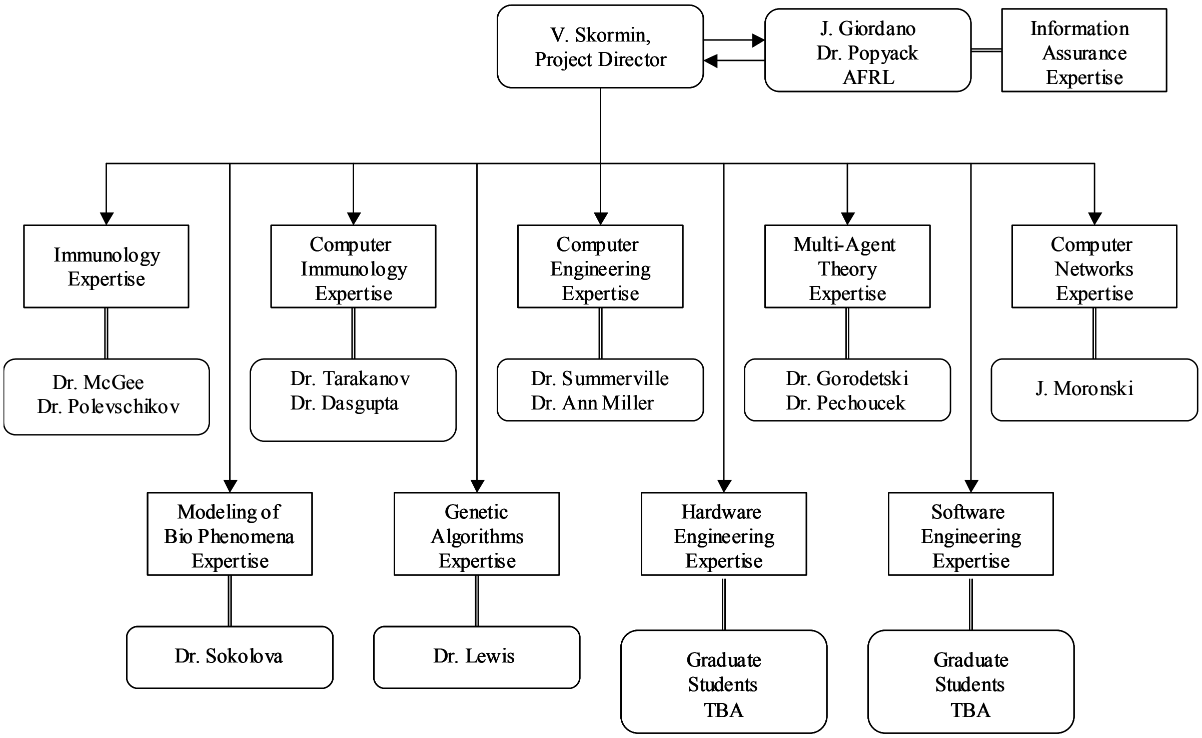
Section VIII describes the experimental computer network testbed that has been developed under the BASIS project. It is suitable for the deployment of various types of computer attacks, including attacks by malicious codes, and conventional and novel information security systems. Equipped with data traffic monitoring and computer forensics systems it facilitates research in the computer network security area and could be recommended as an independent testing site for any newly developed computer defense system.

Section IX of the Report provides the directions for the further research in the BASIS area and provides a brief description of the "Search of the gene of self-replication" project that could be viewed as a direct spin-off the BASIS.

II. Formation of the Comprehensive BASIS Team

The BASIS requires a multidisciplinary effort; the disciplines included are: advanced control theory, mathematics, computer science, computer engineering, biology, information warfare, and computer programming. Therefore, one of the first goals of this project was the formation of a comprehensive, international BASIS team.

Organization of the BASIS Team


V. Skormin, Project Director
J. Giordano Dr. Popyack AFRL
Information Assurance Expertise
Immunology Expertise
Dr. McGee Dr. Polevschikov
Computer Immunology Expertise
Dr. Tarakanov Dr. Dasgupta
Computer Engineering Expertise
Dr. Summerville Dr. Ann Miller
Multi-Agent Theory Expertise
Dr. Gorodetski Dr. Pechoucek
Computer Networks Expertise
J. Moronski
Modeling of Bio Phenomena Expertise
Dr. Sokolova
Genetic Algorithms Expertise
Dr. Lewis
Hardware Engineering Expertise
Graduate Students TBA
Software Engineering Expertise
Graduate Students TBA

The above chart illustrates organization of the BASIS team. It could be seen that this multidisciplinary approach is based on the expertise in

- **Immunology**, provided by *Dennis McGee*, Ph.D., associate professor of the biology department at Binghamton University and *Alexander Polevschikov*, Sc.D., Professor, Senior Research Fellow at the Institute of Experimental Medicine of St. Petersburg Section of the Russian Academy of Sciences

- **Computer immunology**, provided by *Alexander Tarakanov*, Sc.D., Professor, Senior Research Fellow at the Institute of Information and Control of the St. Petersburg Section of the Russian Academy of Sciences and *Dipankar Dasgupta*, Ph.D., Director, Intelligent Systems & Security Research Laboratory, Associate Professor of Computer Engineering at the University of Memphis, TN

- Computer Engineering, provided by *Douglas Summerville*, Ph.D., Assistant Professor of the Department of Electrical and Computer Engineering at Binghamton University and *Ann Miller*, Ph.D., Distinguished Professor of Computer Engineering, University of Missouri-Rolla

- Multi-Agent system theory, provided by *Vladimir Gorodetski*, Sc.D., Professor, Chief Scientist of the Institute of Information and Control of the St. Petersburg Section of the Russian Academy of Sciences and *Michal Pechoucek*, Ph.D., Head of the Agent Technology Group, Assistant Professor of Artificial Intelligence at the Czech Technical University of Prague

- Computer network expertise, provided by *James Moronski*, Chief Engineer at the Endicott Research Group, and computer network consultant

- Modeling of Biological phenomena, provided by *Svetlana Sokolova*, Sc.D., Senior Research Fellow at the Institute of Experimental Medicine of St. Petersburg Section of the Russian Academy of Sciences

- Genetic Algorithms, provided by *Harold Lewis*, Ph.D., Associate Professor of the System Science and Industrial Engineering Department and the Director of the Center for Intelligent Systems at Binghamton University

- Advanced Controls, Applied Statistics, Mathematical Modeling and Simulation, provided by *Victor Skormin*, Ph.D., Principal Investigator, Professor, Electrical and Computer Engineering, Binghamton University

In addition, *Joseph Giordano* and *Dr. Leonard Popyack* of the Air Force Research Laboratory at Rome NY have helped the BASIS team to concentrate their efforts on major information security problems thus addressing the Air Force interests in their research.

III. Attacks on Computer Networks

The problem of information security is recognized as one of the most complex and its importance is growing coherently with increasing network connectivity, size, and implementation of new information technologies. Today, information has become a highly valuable commodity and its vulnerability is of great concern within any large-scale organization utilizing computer networks. Networks and information are becoming increasingly vulnerable to intrusion due to new sophisticated direct and remote threats and attacks. According to a widely accepted point of view *intrusion* is defined as "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource" [1]. According to this definition, there exist three *main types of threats for information security* [2, 3, 4, 5, 8, 9, 10]:

(1) threat of *unauthorized access* to information;
(2) threat of *destroying information integrity*, and
(3) threat of *denial of service* making crucial resource and/or information unavailable.

Denial of Service (DoS) attacks

Denial of Service (DoS) attacks making crucial network resource and/or information unavailable, are among the most common attack types. DoS attacks are both effective and easy to deploy. DoS attacks threaten all systems connected to the Internet, including servers, clients, routers, and firewalls. In a DoS attack scenario, the attacker sends malicious traffic to the target with the aim of crashing, crippling, or jamming communication between the target system and legitimate users.

Fundamentally, there are three DoS attack scenarios, all of which effectively disable the target and prevent its legitimate use. In the first scenario, the attacker exploits bugs in network software implementations, crashing or disabling the target's communication processing capabilities. The second attack type is aimed at weaknesses in network protocols, aiming to overload the system's communication resources, which disconnects the target from the outside world. The third type of attack exploits the limited network bandwidth to the target, inundating it with enormous volumes of traffic.

Clearly, all three forms of attack have the common result of preventing legitimate use of the target system. Although patchwork solutions have been developed for many of the DoS attacks currently identified, new attacks are continually being developed. The purpose of connecting computer systems to a network is to provide access for their legitimate use. Although network protocols can be made more secure, any mechanism that allows outside access to a system can be exploited and makes that system vulnerable to attack. Thus, protecting networked systems requires accepting the dynamic, uncontrollable, and potentially hostile environment in which they exist and developing protection mechanisms that can cope with this environment.

Current computer security systems provide some degree of protection of information attacks and enhance the decision-making ability of the network administrator. These functions are performed by a number of independent system components requiring an enormous amount of distributed and specialized knowledge, and consequently, computations. As a rule, these systems represent a bottleneck with regard to throughput, speed, reliability and flexibility of a network

Information Attacks by the Deployment of Malicious Codes

A large number of information attacks are perpetrated by the development and dissemination of various types of malicious codes such as Trojan horses, worms, and viruses.

These attacks result to the violation of the confidentiality, integrity, availability of the information and, potentially, complete loss of administrative control of the host operation. The main stages of such an attack include implantation of a malicious program into the remote system, execution of the program, information exchange between malicious program and some outside servers resulting in the spread of the attack, gaining control of the attacked host and potentially, the entire network. The particular sequence of such stages depends on type of the attack.

The following are subsets of this class of attacks:

1. *Implantation of a malicious executable file (program) into remote computer systems*. These attacks include implantation and consequent execution of a malicious program in the target system. In addition to rendering the attacked computer useless, malicious programs have a built-in self-replication function resulting in the potential dissemination of this program over the entire community of users and the entire network. Such a malicious program could be activated from a remote terminal or by the initiation of legitimate software that is installed on the target computer.

2. *Implantation of a malicious script or program text into remote computer systems*. This type of attacks has only one feature that makes it different from the above described: it requires that some auxiliary software, such as a script interpreter or a translator for a particular programming language, be installed on the target computer.

The malicious codes include:

1. *Trojan horses*.

A Trojan horse is a program disguised as a legitimate piece of software. Unlike “bugs”, undocumented components introduced by the developers in the legitimate software without malicious intentions, an intruder has to make a special effort to introduce a Trojan horse in the target system through the use of e-mail or in some other way. Very often, a Trojan horse program can self-replicate, which makes it similar to a computer virus. Trojan horses operate differently from viruses, typically facilitating violation of confidentiality (“stealing” passwords) and providing unauthorized access to a computer system (Back Orifice).

2. *Computer viruses*.

Computer virus is a label for several groups of malicious programs intended to violate system integrity and/or cause denial of service. Their main common feature is self-replication. The family of viruses includes: boot-viruses, file viruses, common macro-viruses, viruses on script languages, and worms.

2a) *Boot-viruses*.

Boot-viruses is one of the most dangerous types of viruses for computers utilizing Windows NT/2000 or Unix. They gain control over the computer before the operating system starts and work in real mode. As a result, they can damage not only data and software, but also the hardware.

2b) *File viruses*.

This type of virus infects executable files or shared libraries. When DOS was the most popular operating system for computers with IBM PC AT architecture, file viruses were the most common type of viruses, moving to the second place after macro viruses due to advancements in software.

2c) *Macro viruses and viruses on script languages*.

This type of virus is now the most prevalent due to the fact that many programs utilize their own system tools in the form of script language interpreters. The majority of script languages is rather powerful and can be used for malicious purposes. Different dialects of BASIC (such as VBA or VBScript), responsible for the success of Microsoft software, are commonly used as carriers of macro viruses. The appearance of the virus LOGO in the spring of 2001 provides evidence that practically every language could be used for malicious programs.

3. *Computer worms*.

This type of virus has been known for more than ten years. The first well-known Internet worm was written by Robert Morris in 1988. But the greatest epidemics took place recently: "Melissa" in spring of 1999 and "Love Letter" in spring of 2000. The main characteristic of worms is that they self-replicate and disseminate their copies via computer networks. For example, Melissa and Love Letter used e-mail for that purpose. Internet worms can spread using both executable files and scripts. Some worms consist from several parts: programs, data, and scripts.

4. *Malicious scripts placed into html pages*.

While these attacks are not different from malicious script languages and macro viruses, the Internet browsers provide users with an additional defense mechanism against harmful scripts in html-pages. However, the latest browsers are rather big and complicated programs that can (and do) contain some bugs that could be exploited by intruders.

5. *Malicious JAVA applets*.

There are only few attacks of this class, due to a quite reliable security concept of JAVA known as a "sandbox". Known cases of using JAVA applets for intrusion indicate that these applets have used mistakes in particular realizations of the JAVA machine or the users were careless by utilizing low security level in their browsers for all Internet sites. For example, some JAVA viruses can be “contracted” only when security level in browsers is switched to minimum. The potential problem is that the security level in most browsers is zone-dependent and if malicious JAVA-code gets into a trusted zone it will have a chance to be executed and spread throughout the zone.

6. *Malicious Active-X components (also COM/DCOM/COM+/.Net and -CORBA).*

In this type of attack a user downloads from the Internet an executable code, disregards its digital signature and the certificates proving the validity of that signature, and runs this code without a “sandbox”. In some cases, the use of fraudulent certificates takes place.

In addition, some known attacks may combine several of the above subsets.

Analysis of the attacks with the use of malicious codes indicates that the objective of the attack cannot be achieved to its fullest without the dissemination of the malicious code over the entire community of users, hosts, and potentially the entire network. The attackers skillfully utilize the target computer, before rendering it useless, for expanding the attack onto as many as possible computers at the early stages of the “information terrorism”, utilizing portable media (diskettes, ZIP disks, CD) and now, the power of the Internet.

It is well known that any legitimate software developer invests significant efforts toward protecting his product from unauthorized use. Legitimate businesses may be interested in free dissemination of their advertisement in the electronic format, but being bounded by the rules of legitimacy, are very cautious about the means of achieving this goal. The only type of software developers whose software is intended to automatically spread over as many computers as possible without the users’ consent are the perpetrators of information attacks. Unlimited spread of malicious codes in the process of an information attack is achieved by the introduction of a special self-replication facility in the code, in addition to its main malicious functions.

IV. Mechanisms of the Immune Response

The immune system, like computer network systems, is extremely complex. It comprises a massive whole body response mechanism involving multiple cell types and specialized tissues. For the purpose of feasibility, the immune system needs to be represented/modeled in its basic components and then one could consider the interaction of these components resulting in a complete body response to three generalized foreign agents: a toxin, a bacteria, and a virus. The toxin represents a foreign agent presented to the system in large amounts (e.g. injected or swallowed) or produced in large amounts by an infectious agent that causes harm to the host system. In a computer network environment, effects of a toxin could be paralleled with an illegal entry that results in the destruction of significant amounts of resident data. The bacteria would be an agent which can replicate itself independent of the host and which causes harm to the host system. A corresponding attack on a computer network would comprise an illegal entry facilitating future illegal utilization of the network facilities including the ability to manipulate confidential information. Finally, the virus would represent an agent which replicates itself through the host system and which then would cause harm to the host. Unsurprisingly, growing and replicating itself using the host facilities at the expense of resident software, is exactly what a computer virus does rendering the network useless.

The players in the immune response

Antigens may be composed of several types of compounds (protein, sugars, lipids, etc.), however protein antigens are the type that induces the most vigorous response. Proteins are comprised of chains of simpler compounds called amino acids. Since there are 20 different amino acids, the combinations of the amino acids can yield an incredible number of possible distinctly different proteins.

The antibodies are proteins themselves that, by the nature of their amino acids at the binding sites, can bind strongly to specific short sequences of amino acids. These specific amino acid sequences may then appear within the longer sequence of a certain protein and therefore the antibody can "recognize" the protein via this shorter sequence and then bind. One important property of this antigen-antibody recognition system is that it is extremely specific. However, antibodies may also bind, with lesser strength, to amino acid sequences that are almost identical to the recognized sequence allowing for "cross-reactivity" of the antibodies. This could allow the antibodies to recognize a slight variation of the original infectious agent and therefore confer some immunity. Yet the more different the sequence is from the recognized sequence, the greater the chance for non-recognition. Another important consideration of this system is that in general, a specific antibody must exist for essentially all of the antigens that one could possibly encounter. Still, the immune system has the capacity to randomly generate more than 10^{15} different antigen-binding antibodies.

The antibody proteins are produced by the B cells. These cells randomly generate the capacity to produce an antibody with a single antigen recognition site such that one B cell produces an antibody that recognizes only one antigen (to have a cell which produces several antibodies which recognize different antigens would be a regulation nightmare!). This B cell then produces the antibody only as a cell surface receptor and remains in a resting state, waiting to encounter the specific antigen. When the antigen is encountered, it binds to the antibody on the B cell surface and stimulates the B cell to awake and get ready to function. However, in most instances, the B cell cannot begin to undergo cell division (to amplify the number of antigen specific cells and therefore amplify the response) or begin secreting antibodies until it obtains a second signal from a Helper T lymphocyte (Th cell). This prevents the B cell from producing potentially harmful antibodies without a confirmation that the response is needed. Once the B cell is activated, it then begins to differentiate into either a Plasma Cell, which produces large amounts of antibodies and then dies, or a Memory Cell which eventually reverts back to a resting state and waits for a second encounter with the antigen. These Memory Cells are the basis of the greatly elevated and rapid memory response to the same antigen, as there are now greater numbers of these cells present which now have a less stringent requirement for activation.

As mentioned above, the B cell requires a second signal from a Th cell in order to continue its activation sequence. This Th cell is also an antigen specific cell with a specialized receptor, called the T cell receptor, for a very specific antigen amino acid sequence. The T cell receptor is similar to the antibody molecule, yet it is limited in its ability to recognize and antigen. During development of the Th cell (indeed all T cells including the CTLs and T_{DTH} described below), the cells pass through a specialized tissue, the thymus, in which cells with T cell receptors that recognize self-antigens are killed. This eliminates the majority of the self reactive cells and does an excellent job of preventing autoimmune responses. Also in the thymus, only the T cells with T cell receptors which can recognize antigen segments which are "presented" to them by accessory cells with specialized cell surface antigen-presentation receptors, the Major Histocompatability Complex class II receptors (MHC II), are allowed to survive. Indeed, it has been estimated that greater than 90% of the T cells in the thymus never survive these stringent regulatory requirements to leave the thymus.

Once a Th cell leaves the thymus, it is fully capable of functioning, yet, like the B cell, it too is in a resting state. The T cell receptors of these Th cells cannot recognize antigens alone so they cannot be activated directly by antigen. In the case of the humoral immune response, the B cell binds the antigen via its cell surface antibody that gives the B cell its first activation signal. This antibody bound antigen is then taken internally by the B cell and "processed" into short amino acid segments that are then "loaded" onto the MHC II receptors. The B cell then places these MHC receptors loaded with antigen segments on its surface and is now ready to interact with a Th cell. This interaction then consists of the B cell "presenting" the antigen to the Th cell to activate the Th cell. This presentation of the antigen-MHC II to the Th cell an activation signal for the Th cell to begin cell division (amplification of the response) and to differentiate into a mature helper T cell capable of helping the B cell. The mature, activated Th cell then produces signals (via cell surface receptors or small secreted factors) that tell the B cell to continue on its activation sequence to cell division and antibody secretion.

Of interest, this B cell-Th cell interaction presents another site for amplification of the immune response. One B cell can "process" a large antigen into several small different segments that could be used to activate several Th cells with different antigen specificities. This would increase the probability that the B cell would get a second signal from a Th cell even if the Th cell did not recognize the exact same segment of the antigen amino acid sequence that the B cell recognized. Also, a single activated Th cell could then interact with several B cells to allow the production of several different types of antibodies (one specific type from each different B cell). However, only those B cells which have encountered the antigen for the first activation step would be sensitive to the Th cell help. The overall response would be a more complete activation of several B cells and T cells with different antigen specificities - essentially responding to several different segments of a single antigen.

Before continuing, another important group of cells must be considered. These are the accessory cells that are not antigen specific but play a very important role in the immune response. The accessory cells consist mainly of macrophages and dendritic cells which function to engulf or phagocytize cells, bacteria, viruses, or even cellular debris and proteins. These engulfed cells or substances are then enzymatically destroyed and "processed" much like the B cell processes antigens bound to their cell surface antibody receptors to yield short segments of amino acids. As in the B cells, these antigen segments are also loaded onto MHC II receptors for the accessory cells to "present" to any nearby Th cells. Indeed, the initial activation of most Th cells usually occurs via the presentation of foreign antigens by these accessory cells. This system then allows for the constant sampling of the body's environment via macrophage phagocytosis (for bacteria, viruses, and etc.) or dendritic cells (for cell debris and individual proteins). Therefore, antigen sampling (and hence Th cell activation) is not limited to antigen specific B cells only, but also certain non-specific accessory cells.

In the cell-mediated immune response, one of the most potent killer cells is the CTL, a T cell with a T cell receptor that recognizes foreign antigens present inside of an infected cell. Like the B cell, this CTL is normally in a resting state and needs two independent signals for activation. The first signal comes when the resting CTL encounters an infected cell. Almost all cells of the body have an internal system that constantly destroys old proteins as new ones are produced by the cell. The destruction of old proteins results in the production of short amino acid segments which may then be loaded onto a different type of MHC receptor called the MHC class I receptor (MHC I) which is then placed on the surface of the cell. Therefore, most cells of the body constantly display a variety of "self" amino acid segments in conjunction with the MHC I receptor on the surface of the cell. However, when a cell becomes infected with a virus, the virus uses the cells machinery to replicate itself. Yet this replication of the virus inside of the cell allows the cell's internal system to sample some of the virus proteins by destroying them and placing the short virus amino acid segments onto the MHC I receptors (again, random sampling as with the accessory cells above). Subsequently, when the viral antigen loaded MHC I receptors are on the surface of the cell, the cell is now labeled as an infected cell even though the immune system cannot directly get at the virus inside of the cell.

Once a CTL comes in contact with a virus infected cell, if its T cell receptor can recognize the virus segment, then the CTL obtains its first activation signal. However, the CTL cannot be completely activated until it receives help from a Th cell which has also been activated. The activated Th cell (usually activated by antigens from the same virus as presented by accessory cells) then produces a T cell growth factor (interleukin-2) necessary for the CTL to begin cell division (once again, an amplification step) and mature into a functional CTL. As with the B cells, memory CTLs are also produced to produce a memory response in subsequent encounters with the virus. When the mature CTL encounters the virus infected cell again (via the T cell receptor binding to the virus antigen segment and the MHC I receptor), the CTL kills the virus infected cell. Of importance, the mature CTL can kill many virus infected cells over its life span.

The final player in the cell-mediated immune response is the T_{DTH} cell. Once again, this T cell has an antigen specific T cell receptor and must have the antigen presented by an accessory cell along with MHC II. They are essentially helper T cells that have their first encounter with the antigen in the lymph nodes. They then undergo cell division (amplification of the response) and maturation to competent T_{DTH} cells. These cells then leave the lymph nodes and actively seek out the areas of infection (see below how they are recruited into areas of infection). They migrate into the area where they receive the second encounter with the antigen (presented by resident macrophages actively phagocytizing the bacteria or viruses or dendritic cells sampling the infection debris) and then produce large amounts of inflammation inducing factors or cytokines which activate the phagocytic macrophages, neutrophils and other cells in the area.

The interaction of the immune system components against toxins

In the case of a toxin, the best defense is the antibody molecule. A toxin by itself usually can cause harm to the cells of the body and therefore must be neutralized. This is usually effectively done by the binding of the antibody molecule (usually by blocking the toxin from entering into a cell or blocking the toxin function). Therefore, the major players in the anti-toxin response would be the B cell that produces an antibody to neutralize the toxin and the Th cell that provides help for the B cells. However, macrophages can also have a role in these responses by providing another cell type to present the toxin antigens to Th cells. Macrophages also have receptors on their cell surface that can bind to antibodies that have bound to an antigen (these receptors often do not bind well to antibody which has not bound to an antigen). This then provides a way for the macrophage to attach to the antigen and engulf or phagocytize the toxin to remove it from the body. Finally, some toxins may also activate macrophages and induce them to secrete soluble factors that can enhance B cell division, antibody production, and Th cell responses. On the second encounter with the toxin, the body already has antibody present to neutralize the toxin and the greater number of antigen specific B cells and Th cells (memory cells) are rapidly activated to produce extremely high levels of anti-toxin antibody (which is the basis of the booster shots in vaccines).

The interaction of the immune system components against bacteria

For a bacteria not sequestered inside of a cell, the first line of defense is the innate immune response: the inflammatory response and macrophage phagocytosis of the bacteria. The purpose of this innate immune response is to hold the infection at bay until the immune response can be activated. Macrophages which have phagocytized bacteria or dendritic cells which have picked up bacterial debris begin to present bacterial antigen segments with MHC II. These cells travel to nearby lymph nodes where they then can present the antigens to the Th cells to begin their activation. Meanwhile, bacterial debris or even whole bacteria present in the lymph (the fluid surrounding the cells of the body) are carried via the lymphatic system to the lymph nodes. This allows for B cells in the lymph nodes to become stimulated and even resident macrophages in the lymph nodes to pick up antigen for presentation to Th cells. The activated Th cells then interact with the activated B cells and eventually the B cells begin to produce massive levels of antibody. This antibody then gets into the blood circulatory system and is carried to the infection site where it can have a number of effects. Antibody binding directly to bacteria can allow macrophages and neutrophils to attach to the antibody to enhance phagocytosis and killing of the bacteria. Antibody bound to the bacteria can also activate the inflammatory system that eventually results in the activation of macrophages to become better bacteria killers and to cause the release of signals which recruit more macrophages, neutrophils, and even T cells from the blood to the site of infection.

In some instances, T_{DTH} cells in the lymph nodes may also be activated by the accessory cells bringing in antigen. These T_{DTH} cells then leave the lymph nodes to seek out the area of infection. Once in the infection area, the macrophages present more antigens to the T_{DTH} cells to induce them to release several powerful inflammation inducing factors. These include factors that recruit more macrophages and neutrophils from the blood into the area, factors that activate the neutrophils and macrophages to become master killers of microbes (this in addition to the virus-specific antibody greatly enhances the macrophage function), factors that provide help for other Th and T_{DTH} cells in the area, and they can help B cell to enhance antibody production. The end result of these responses is a massive influx and activation of killer macrophages and neutrophils which phagocytize the bacteria, the influx of antibodies which neutralize the bacteria and enhance their phagocytosis, and the activation of the inflammatory response. The invading bacteria are usually destroyed, however host tissue damage may occur in cases of massive infection. Of note, often the induction of the immune and inflammatory response results in the secretion of high levels of activation factors by the macrophages and T cells. As the levels of these activation factors increase, they often induce the production of inflammation suppressing factors by the immune cells and resident cells of the tissues. This, along with the reduction in the levels of antigens or bacteria for stimulation, allows for the down-regulation of the response and the beginning of wound healing.

The interaction of the immune system components against viruses

Viruses present an interesting challenge to the immune response in that these agents have an intracellular phase, in which they are not available to many of the immune response elements, and often an extracellular phase when the virus is shed from an infected cell to spread to and infect nearby cells. Most of the above immune mechanisms (antibodies and T_{DTH}-macrophage responses) can effectively handle the extracellular phase of the virus. Antibodies bind to the extracellular viruses and prevent their binding to or entering other cells, enhance their destruction by allowing macrophages a handle to bind to the bound antibody and induce phagocytosis, and bound antibody can induce the inflammatory response. T_{DTH} type helper T cells can migrate to the site of infection and direct the activation of macrophages to become master killers with a greater phagocytic capacity, produce factors or cytokines which recruit other T cells, macrophages, and neutrophils into the area of infection, help the activation and maturation of CTLs (see below), and, since the T_{DTH} cells are specialized Th cells, they can help B cells to enhance antibody production. In addition, the T_{DTH} cells can secrete a very potent factor, interferon, which induces all nearby cells to turn on their own internal antiviral defense mechanisms to prevent viral replication and help in preventing the spread of the infection.

Yet the above mechanisms generally have no effect on the viruses hidden within infected cells. The result is that the infection continues because the source (the virus infected cell) has not been destroyed and in some cases the infection can spread via direct cell-to-cell transfer of the virus without an extracellular phase. The destruction of the virus infected cells requires the action of the antigen specific CTLs. CTLs activated at the site of the virus infection can receive immediate help from the T_{DTH} type Th cells in the area to become mature, active CTLs to kill the virus infected cells. This also releases any internal viruses to be exposed to antibody and macrophages for destruction. Finally, CTLs also can produce interferon which induces more nearby cells to turn on their internal antiviral mechanisms.

The overall effect is that the virus spread and source of infection is stopped. Of course large numbers of memory B cells, T_{DTH} type Th cells, and CTLs are also produced so that in subsequent encounters with the same virus, the specific immune response is very rapid and much stronger; hence, immunity.

Whole body responses to infections

In addition to the above described immune responses to infectious agents, several other mechanisms are induced which can help in preventing the spread of the infectious agents to different parts of the body.

One of the most striking features of the immune system is that the immune response cells are not centralized, but are spread out in strategically placed lymph nodes throughout the body. The fluids collected from around the cells in only a defined section of the body pass through any single lymph node (e.g. the lymph nodes of the groin area filter fluids from various sections of the legs). These lymph nodes provide a staging area for the interactions that are required for the immune response to occur, interactions that could not occur in the rapidly flowing blood or most normal tissues where the immune cell numbers would be too low. To ensure that the antigens of the infectious agents get to the lymph nodes (often well before the antigens or viruses and bacteria actually reach the lymph node on their own), the macrophages and dendritic cells at the infection site specifically migrate to the local nodes carrying samples of any infection in the tissues. This way, the immune system does not have to initially seek out the infection - it is brought to the immune system. The lymph nodes also provide a filter where lymph node macrophages remove many of the bacteria and viruses from the fluids to prevent the spread of the infection. Indeed, several lymph nodes may be strung in succession to ensure the filtering of infectious agents. Therefore, the immune response is localized and direct for a specific area of the body.

However, the results of the localized lymph node immune response are disseminated throughout the body. Antibodies and infection-seeking activated T_{DTH} cells and CTLs quickly reach the blood circulatory system and are spread throughout the body to prevent the spread of the infection. As mentioned above, these cells are actively recruited to the areas of infection by the factors produced as a result of the inflammatory response and activated immune cells. After the close of the immune response, the memory B and T cells continue to migrate throughout the body, spending varying amounts of time in each lymph node on the way. This insures that the memory cells will then be (or soon will be) at the appropriate lymph node to respond to a second encounter with the antigen wherever it may occur.

V. BASIS: Immune and Computer Defenses

As computer networks become increasingly vulnerable to cyber terrorism, an attempt to develop a proactive defense mechanism inevitably steers designers towards already existing, naturally evolved, biological defense mechanisms. Biological immune systems are adaptable, self-regulating, automatic natural defense mechanisms that work to protect vertebrates from disease. Using the immune system as a model, a system to protect computers and computer networking systems from information attacks can be created. In order to build this model, it is first necessary to understand the similarities between a biological system and a computer networking system. In addition to analyzing immune systems, attack mechanisms will also be investigated. Here, the Denial of Service (DoS) attack will be studied.

The Components of a Biological Immune System

The immune system's function can be broken up into two functional units, innate immunity and acquired immunity. Innate immunity, also referred to as non-specific immunity, forms the basic resistance to infection that is possessed by any given species. It is comprised of many different types of barriers that are the first line of defense which protect the body from infection. These are the anatomic barrier, physiologic barrier, endocytic and phagocytic barriers, and the inflammatory response. Acquired immunity, additionally known as specific immunity, is the active portion of the immune system that is capable of recognizing and eliminating antigens. Acquired immunity is a much more complex system than innate immunity and is composed of specialized cells responsible for the detection and destruction of pathogens.

The anatomic barrier is comprised of the skin and other epithelial surfaces (stomach, intestine, respiratory, urinary tracts and others). These are the interfaces between the body and the outside world and work to effectively stop antigens before they are able to enter the body and start infection or cause damage. While the anatomic barrier participates in the immune system, it generally must be told what to do. Problems such as breaks in the epithelium, microbial infection, and damage to the tissues are detected and initiate the inflammatory response, the body's first line of defense against infection that includes the activation of phagocytic cells.

Physiologic barriers, working with the anatomic barrier further decrease the risk of infection. Temperature and pH are examples of physiologic barriers. When a pathogen comes in contact with, or passes by, the skin or epithelial surface, the environment that is present can slow or potentially prevent its growth and spread. Physiologic barriers provide some explanation as to why different animals are naturally immune to certain diseases. For example, chickens are naturally immune to anthrax because their high body temperature prevents growth and infection.

A more complex innate immune behavior is found in the endocytic and phagocytic barrier. Here, extracellular materials are ingested through endocytosis or phagocytosis. Ingestion occurs as certain cells internalize, by folding their cell walls around, the extracellular material forming endocytic vesicles. Upon internalization, these vesicles are routed to endosomes where the process of dissociation of the extracellular material begins. This process is assisted by the inflammatory response.

The inflammatory response is characterized by an increase in the diameter of the blood vessels (vasodilatation), increased capillary dilation and an increase of the presence of phagocytic cells. Inflammation occurs as a result of a complex interaction with a variety of chemical mediators. Vasodilatation and increased capillary dilation allow for white blood cells, which are composed mostly of phagocytic cells, to migrate to the affected area. In addition, inflammation serves to contain any potential infection to a localized area.

When antigens breech the innate defense systems, acquired immunity is the next system involved in elimination of specific foreign microorganisms and molecules. The acquired immune system displays specificity, memory, diversity, and self/non-self recognition [1]. Acquired immunity however does not act independently from the innate system, but in concert with it. Innate responses trigger responses from the acquired immune system. For example, soluble proteins released during the inflammatory response attract cells of the acquired immune system.

The acquired immune system is composed of two groups of cells called lymphocytes and antigen-presenting cells. Lymphocytes are one of many types of white blood cells that are produced in the bone marrow and circulate in the blood and lymph system. Lymphocytes are responsible for the specificity, memory, diversity, and self/non-self recognition found in the immune system. B-lymphocytes and T-lymphocytes comprise the majority of the lymphocytes.

B-lymphocytes, upon maturity, leave the bone marrow with a unique antigen-bonding receptor, which is an antibody, on their membrane. When the B-lymphocyte first encounters an antigen that matches its receptor, it begins to divide rapidly. Both memory B cells and effector cells called plasma cells are created. The memory B cells that are created retain the antigen-bonding receptor on their membranes and function as B-lymphocytes. The plasma cells do not form an antigen-bonding receptor. Instead, the antibody is secreted and acts as an effector molecule.

T-lymphocytes, unlike B-lymphocytes, mature in the thymus gland. Like B cells, T cells have an antigen-binding receptor on their membrane. Unlike the B cells, the T cell receptor cannot bind with an antigen unless the antigen is also associated with cell membrane proteins known as major histocompatibility complex (MHC) molecules. When a T cell comes in contact with an antigen associated with an MHC molecule, it begins to divide forming other effector T cells.

The lymph nodes are the home of many of the immune cells of the body. T cells act as the directors of the immune system. B cells produce antibodies, which bind to foreign antigens, and cytotoxic T cells kill virus infected cells. Amongst the immune cells are the antigen presenting cells (macrophages and dendritic cells) which sample antigen throughout the body, process them, transport them to the lymph nodes and present the processed antigen to the T cells. The sampling of antigen is continuous, including the sampling of self, for which there are no T cells to bind with, thus preventing a self-attack.

The Components of Computer Networking Systems

Computer systems and networks of computer systems already posses many of the characteristics of the immune system. In computer systems, anatomic barriers exist in the form of simple username/password security, connection encryption and process/user privileges. Extended to computer networks, hubs, switches, gateways and routers in their basic form provide effective barriers against many types of potential attacks. Furthermore, these systems can provide added security when an attack is underway by blocking or shunting potentially harmful network traffic away from sensitive systems or networks. The diversity of computer systems and networks also form another barrier to attack. Security holes and exploits usually affect specific operating systems on specific hardware platforms.

Antivirus and intrusion detection packages are created to further protect systems from infection and damage. One could almost consider this the “active” part of the computer’s or computer network's immune system, but, in reality, these are nothing more than "dumb" barriers that are sometimes updated, usually after infection, after damage has already been done. Furthermore, no one tool has been developed to solve the problems with computer systems infection as infections can occur across platform, with different operating systems and different network implementations.

The Denial of Service (DoS) Attack

The Denial of Service (DoS) attack is the most threatening type of attack that can be launched against a computer or network of computers. The DoS attack endangers all computers systems connected to the Internet, including servers, clients (workstations), routers, and firewalls. In a DoS attack scenario, the attacker sends malicious traffic to the target with the aim of crashing, crippling, or jamming communication between the target system and legitimate users, effectively "killing" the computer system or the services that it offers.

Fundamentally, there are three DoS attack scenarios, all of which effectively disable the target and prevent its legitimate use. In the first scenario, the attacker exploits bugs in network software implementations, crashing or disabling the target's communication processing capabilities. The second attack type is aimed at weaknesses in network protocols, aiming to overload the system's communication resources, which virtually disconnects the target from the outside world. The third type of attack exploits the limited network bandwidth to the target, inundating it with enormous volumes of traffic.

Clearly, all three forms of attack have the common result of preventing legitimate use of the target system. Although patchwork solutions have been developed for many of the DoS attacks currently identified, new attacks are continually being developed. The purpose of connecting computer systems to a network is to provide access for their legitimate use. Although network protocols can be made more secure, any mechanism that allows outside access to a system can be exploited and makes that system vulnerable to attack. Thus, protecting networked systems requires accepting the dynamic, uncontrollable, and potentially hostile environment in which they exist and developing protection mechanisms that can cope with this environment.

The three primary DoS attack scenarios have common characteristics as well as differences in their implementation and execution. The main characteristic shared by all DoS attacks is the basic aim of preventing normal access to the system under attack. The main differences lie in the degree of infiltration of the system under attack and the duration and severity of the damage inflicted.

DoS attacks that exploit bugs or weaknesses in network software implementations are designed to cripple the target by crashing or modifying the networking software or the operating system itself. Generally, the attack can be launched from a single computer by sending malformed packets or sequences of packets to the target. This scenario has the highest level of infiltration; malicious network traffic enters the target system and cripples it. The damage inflicted on the target is severe and persists until the affected software is re-installed. Since these attacks are due not to weaknesses in the network protocols but their implementations, they are easily fixed once the bug is identified and the proper software patch applied. However, the complexity of network software implementations often makes it impossible to test for and eliminate all bugs, making them vulnerable to future attacks.

The second form of DoS attacks directly target weaknesses in the high-level network protocols. The objective of these attacks is not to crash the target but to cripple it by overloading a bottleneck resource. In this scenario, the attack is aimed at the entry points to the system and represents a medium level of infiltration. No damage is inflicted on the system software (it is functioning normally). In addition, access to the target computer may not be completely blocked off by the attack and usually resumes shortly after the attack ceases. The most prevalent example of this type of attack is the SYN Flood or TCP SYN Attack. Like software implementation bug attacks, mechanisms can be put into place to prevent attacks against network protocols once a specific attack scenario is identified. However, these protocols are complex and likely to have further vulnerabilities.

The first two forms of DoS attacks, those against network protocols and their implementations, can be launched from a single source. The third class of DoS attacks represents coordinated efforts launched against a target from multiple, distributed sites. These attacks can be particularly troublesome as they may invade the target network from different points of entry, allowing an increase in the volume of malicious traffic that can reach the target. The goal of this form of attack is to jam the communication pathways to the target without disabling the processing on the target system itself. In this scenario, the attacker doesn't infiltrate the target system, but effectively jams the access paths to it. Access to the target is not completely prevented during the attack (although it is severely limited) and access resumes quickly after the attack ceases. This specific class of attacks, known as Distributed Denial of Service (DDoS) attacks, rely on recruiting a number of zombie computers that are subjugated by a master to assault the target from a number of locations. Defending against a DDoS attack requires a coordinated, distributed defense.

Anatomy of a DoS TCP SYN Flood (TCP SYN) attack

TCP (transmission control protocol) is one of the more common protocols used by machines to communicate across networks and is defined in[RFC 793]. TCP provides a reliable ordered connection for the exchange of data. In order to establish a connection, some configuration data needs to be transferred between client and server. This is done using a three-way handshake.

The client first informs the server of its beginning sequence number. This is done with the SYN bit set in the TCP header; later referred to as the SYN packet. The server receives the SYN packet, "half-opens" the connection and acknowledges the received SYN while also sending its beginning sequence number, this is known as the SYN-ACK packet. At this point, the server is in the SYN-RECEIVED state. When the client receives the SYN-ACK, it responds to the server with an ACK to acknowledge the server's beginning sequence number. After reception of the final ACK by the server, the connection is fully established. For every unique SYN packet (the host address, port, and sequence numbers are the same), the server creates a half-open connection and responds with a SYN-ACK.

Another important feature is that the network is considered inherently "unreliable" hence network conditions may cause the loss of the SYN, SYN-ACK or ACK packet. If the client does not receive the SYN-ACK within a reasonable time, it will continue to retransmit the original SYN request until a connection timeout is reached. If the server does not receive an ACK to the transmitted SYN-ACK, it will continue to retransmit the SYN-ACK until an ACK is received or until some predetermined SYN-ACK timeout is reached upon which the connection will be closed and previously allocated resources will be released.

A malicious client takes advantage of the fact that when the server side of the connection is in the SYN-RECEIVED state, the server has allocated network resources for the "half-open" connection. When the client sends the initial SYN packet, it usually "spoofs" the client's address.

This is done for two reasons. First, the actual client launching the attack is disguised by the spoofed address. Second, when multiple SYN requests are sent out with different source addresses, the attack will look as though multiple hosts are requesting connections. Once the server's available networking resources are consumed, the server becomes effectively shut off from the world.

Since only one client is required to launch a multi-pronged assault on a target server, the SYN Flood attack is one of the easiest attacks to implement. The bandwidth requirements for an attack of this nature are relatively small. This allows for devastating attacks to be launched from bandwidth-limited connections. Tools to launch an attack are readily available to a wide variety of attackers, from "script kiddies" to professionals. Even though this type of assault is well known, the protocol cannot be changed to prevent an attack without breaking millions of client implementations. Furthermore, it takes only a small step to convert a single machine assault to a distributed and coordinated assault on an entire network.

Once the target system or network is under attack it is very hard to stop the assault. If the assault is determined to be from an application with a hard-coded set of target addresses, then the fix is relatively easy: a simple change of the host address of the targeted system and updating the DNS (domain name service) records for the target will do the job. However, if the attack includes either resolving the new location of the target or moving from source machine to source machine, the level of work required to stop the assault grows enormously.

Until the flow of invalid SYN packets is stopped at the source, the network must have some level of filtering implemented. Since these SYN packets look like legitimate traffic, the source address would have to be verified for each connection request. There are many types of routers and bridges that can assist in performing this action, but network throughput drops when each and every request needs to be verified. Furthermore, many man-hours are wasted with network administrators working towards blocking all of the offending source addresses and networks.

While the packet filtering operations are being put into place, the next step would be to find the actual source of the attack. This process could take days or weeks. With the large levels of administration and bureaucracy between the attacker and the target, the source may never be found or stopped. In fact, the attacker may not be in the same political jurisdiction as the target. The attacker can also halt the attack as an attempt to prevent detection, only to restart it at a later time.

Anatomy of a Relevant Biological Attack

Infectious agents can enter a body in many ways. The consumption of food, air, and water required for life also constitute sources of infectious agents. Tissue damage, such as scrapes and cuts, provides additional paths for possible infection. The anatomic barriers prevent most sources of infection by making the environment inhospitable for infectious agents or just preventing their entry into the body. However, when an antigen makes is past these anatomic barriers, the active parts of the immune system work in concert with the anatomic barriers to first detect, then fight off infection.

The variety of agents includes bacteria, viruses, and foreign proteins. The human body presents a hospitable environment for the development, adaptation and consequent proliferation of the antigens. In return for this hospitality, bacteria can multiply at an alarming rate, releasing toxins and competing with nutrients in the host. For example, the Anthrax bacterium, (B. anthracis), is contracted by the host through inhalation, ingestion, or direct contact with contaminated material. Anthrax releases a series of proteins which kill the host's own cells in the infected area. Untreated, Anthrax can potentially kill the host.

One of the first responses of the immune system is the inflammation response. Inflammation occurs as a result of a release of chemical moderators from damaged cells. The cellular damage can be from trauma, temperature, contact with foreign substances and a host of other reasons. With the presence of the chemical moderators the local blood vessels dilate (vasodilatation). When this occurs, liquid fills in the affected region and it swells. This works to prevent contaminants from spreading beyond the affected region, in effect, isolating the damage. Furthermore, cells from the active portion of the immune system are attracted to inflamed areas and flow outside of cellular walls into the affected areas.

Macrophages are constantly working by picking up extra cellular material, processing it and transporting it the lymph nodes. In the lymph nodes, the macrophages will present the material to the T cells. If a T cell is present that recognizes the material, it (with potentially several T cells) will activate. The activated T cell will then undergo cell division many times to increase the number of cells which can respond to the infection (amplification of the response) and then these cells will begin to differentiate into effector T cells which will actually do the response.

The effector T cell has a fixed lifetime of two to five days, after which it will automatically die. This allows for the proper regulation of the response to ensure that the response does not go on longer than needed. The selected effector T cell then goes to the epithelium and directs the execution of the protection. Effector T cells can secrete soluble factors (cytokines) to tell the epithelium to do things like express new receptors, produce new cytokines, open or close passages between epithelial cells, etc. They also could tell the tissue macrophages to become better antigen presenting cells and become better phagocytic cells. Finally, the effector T cells may be of a killer type (cytotoxic T lymphocytes or CTLs), which seek out virus-infected cells and kill them.

The nature of the infection and the type of cells activated would then determine the effector response generated such that the response is fine tuned to the infection. The immune response continues as long as there are functional effector T cells and that there is a need for the response. As long as there is antigen present, more effector T cells will be generated to replace the ones that are dying because of their fixed lifetime. This ensures that the immune response is maintained as long as needed. Once the antigen level begins to fall, there are fewer stimuli to produce new effector T cells. The remaining effector T cells simply die off at their prescribed lifetime.

Immune system memory is also developed during this process. As effector T cells are created, memory B cells, specific to the attacking antigen are also created. As more effector cells are created, the larger the infection, more memory cells are created. The memory B cells accumulate and have a much longer lifetime of ten to twenty years. Therefore, any second encounter with the antigen or pathogen will trigger the immune response more quickly, unlike the first encounter with the antigen or pathogen where there were only a few B cells with could respond to the antigen. If the body is continuously exposed to the antigen throughout its life, memory cells continuously are produced, therefore making the memory permanent in the body.

Similarity Between Biological Systems and Computer Network Systems

In order to extend the immune system model as a system to protect against DoS attacks to a network of computer systems, similarities between DoS attacks and biological attacks need to be established.

Computer Network	Biological System
The attacks are *very easy* to implement and launch. There are publicly available tools that can be used to launch DoS attacks, requiring only a basic amount of computer programming knowledge	The attacks are essentially ubiquitous, with some type of pathogen potentially existing in almost every environment
It is not possible to prevent or preempt or stop a DoS attack because the attackers most likely lie outside of the administrative control of the system to be protected	It is not possible to control the source of the infection, as the body must take in large quantities of various materials needed for survival
Networked systems must remain accessible and open. A networked system must accept packets from the network	Material must be consumed for survival of the life form
Network packets may contain encrypted data; thus it may not be possible to detect an attack until after it has begun	The infectious agents are often "hidden" within the volume of materials taken into the body. All of the consumed material is treated as beneficial
Once launched, the attack may not be detectable until it has already done significant damage	The host may not appear ill until several hours or even days later (the incubation period for infection) when the infection is well established
Stopgap measures are only minimally effective. Patches can be applied to fix bugs in software implementations and protocols can be made more secure. However, it can be argued that no software is immune to bugs and thus network software implementations will always be vulnerable	Modern environmental health and sanitation laws and practices along with epidemiological studies have helped to control the sources of some common pathogens, however it is not possible to completely prevent infection
Information security systems are modular by nature and comprised of independent software and hardware modules such as username/password facilities, public/private key authentication systems, antiviral software, cryptographic devices and intrusion detection hardware/software	Immune systems are modular by nature comprising such elements as skin, etc…

Differences Between Biological Systems and Computer Network Systems

It could be seen that functions of both systems are similar. Both systems have a complex configuration, however, the biological immune system is more reliable and dependable. Partially, this could be explained by the fact that the "immune problem" is much simpler: the immune system has to differentiate only self from non-self. With this ability, potential attacks can be thwarted before any damage is done. The equivalent in the computer systems world would be the ability to determine normal and abnormal network traffic and malicious software. Abnormal network traffic and malicious software, when discovered should be ignored or blocked.

The biological system also possesses a variety of systems that offer early warning that something is wrong. For example, the inflammation response occurs upon damage, without the presence of an infectious agent. This type of response works to limit the damage that could possibly be done if there are any infectious agents. In a DoS attack against a computer network, by default, all networking components work to "assist" the attack. All of the networking components pass along the malicious packets without any regard to source or content as long as the packets meet the minimum network protocol specifications. The target server even responds as though these packets are legitimate traffic.

In a computer security system, usually only one facility is available for the response to any specific type of attack. The immune system is composed of many components, some layered and some independent, any of which can be inactive, disabled, or defeated without preventing a successful immune response.

The immune system components also communicate with each other in an ad-hoc measure. Most communication is a simple interaction with chemical moderators that are often released to initiate a further response pending presence of other immune system components. Once the response is underway, the responders also release moderators that control the rate of both the number of responders and the size of the response. In computer networks, there is little or no coordination between the network safeguarding systems. Often, the system administrator intervention is the reacting mechanism. Unfortunately, this happens long after an attack begins and perhaps even after major damage has been sustained.

Since the immune system is composed of small independent functional blocks, the resources of the immune system are scalable with many functions working in parallel to fight off an attack and prevent further assaults. If there is an infection, once detected, many of the immune system components can increase their presence by multiplying, with many actively working to suppress the immediate attack and some remaining to fight off a recurrence. In the computer network, often each system is isolated, working independently, with a few pieces of software running to prevent or detect an attack. Each piece of software added adds processing load to the system, decreasing performance. Typically, the response to an attack on the computer network is either all or nothing. There is no scalability depending on the size or success of the attack.

Finally, the immune system is completely automatic. Little "user intervention" is required for it to successfully prevent or eliminate an attack against the host. The immune response scales to the size of the perceived attack and properly scales back when the assault ends. An attack against a computer network often involves direct intervention by the administrators of the network. Many man-hours are spent in the detection, prevention, and response to assaults against these networks. Often computer attacks are successful from complacency, poor training, lack of attention to software holes and patches, and plain incompetence.

Dynamics of the Immune Response

A quantitative description of the autonomic immune response provides a means for understanding major interactions between the components of the immune system and the intruding antigen, and the dynamics of these interactions. Moreover, it allows for the establishing the conditions for three possible outcomes of such interaction, full recovery from infection, chronic infection, and death of the host. It should be clear that the proposed equations provide a simplified description of the actual biological phenomena. Moreover, these equations are only as accurate as their constants. In the case of the immune system these constants can be only roughly guessed. Therefore, the established equations cannot be effectively used for the prediction of the outcome of a disease caused by the antigen or for the modification of the immune "control law". However, the importance of these equations for the computer network applications shall not be underestimated.

First, they describe the principle of operation of a system providing very successful defense against information attacks. Second, because in the case of a computer network, parameters of the equations could be accurately estimated that makes them accurate and dependable. Third, they could be instrumental for the analysis and design of defense mechanisms intended for computer networks.

The specific immune response is the main mechanism enabling the immune system to destroy intruding antigen. Various lymphocytes and proteins that are uniquely equipped for counteracting any particular antigen accomplish this. Each lymphocyte is keyed with a specific receptor for specific non-self proteins. The immune system continuously manufactures these receptors, specializing to at least 10^{20} various genotypes.

The actual ability of the immune system to destroy the intruding antigen depends on the concentration of corresponding specialized fighter cells and antigen. If a specific antigen has not attacked the host, the concentration of fighter cells responsible for a respond to an attack from the particular antigen are minimal. Since it is unlikely that 10^{20} different antigens will attack the host in any particular period in time, an immune memory is developed to counteract common assaults.

During a response to a specific attack, two types of immune cells are created. To assist in the current assault, short-lived fighter cells are created. For potential future assaults, long term "memory cells" are created with a lifetime of ten to twenty years. These memory cells increase the natural background level of the specific antigen responsible for the assault thereby increasing the probability of the detection of an attack in any period of time.

Consider the immune response in the case when a specific antigen infects a biological organism. Introduce quantity C(t) representing the concentration of the immune cells carrying the genotype of the intruder and, therefore, uniquely specified in counteracting it. This concentration is time-dependent, and in the absence of the corresponding antigen is subject to a slow natural decrease that could be expressed as a slow decaying exponential process

$$C(t) = C(t_0)e^{-\alpha(t-t_0)} \tag{1}$$

where

t – is current time,

t_0 - is some fixed moment of time, $t_0 < t$,

α - is a positive constant that could be chosen to reflect the following reality: during a 50-year period the concentration decreases to less than one percent of its initial value.

It is more practical to rewrite the above equation as

$$C(t) = C_0 e^{-\alpha t} \tag{2}$$

where $C(t_0) = C_0$. Equation (2) should be considered as the "natural motion" equation of the immune defense mechanism. It could be seen that the natural dynamics of the immune mechanism is *stable*.

Introduce quantity P(t) representing the concentration of the intruding antigen cells. This concentration is also time-dependent: the antigen utilize the available resources of the biological organism to multiply, and in the absence of the corresponding fighter cells this concentration is subject to a relatively fast increase that could be expressed as

$$P(t) = P(t_1)e^{\beta(t-t_1)} \tag{3}$$

where
t_1 - is the moment of infection, $t_1 < t$,
β - is a positive number, $\beta >> \alpha$

Equation (3) represents the antigen multiplication process within a biological organism in the absence of the immune response attempting to counteract this process. Therefore (3) should be considered as the “natural motion” equation of the antigen multiplication process. It could be seen that the natural dynamics of this process is *unstable*. It should be realized that the antigen multiplication takes place at the expense of the invaded biological organism and therefore quantity β could be viewed as the “available share of the resources” of the organism and is time-dependent.

The immune response is preceded by the detection of the invaded antigen. This is accomplished by the physical contact between the antigen cell and the specialized immune fighter cell that takes place some time after the moment of infection, t_1. Consider the moment of the detection of the antigen, $t_2=t_1+\tau$. While both the antigen and fighter cells are dispersed within the body, the time delay is a random variable and could be characterized by the its average value that is inversely proportioned to the concentrations of these cells, C(t) and P(t), i.e.

$$\tau = \frac{k}{C(t)P(t)} \tag{4}$$

where
$k > 0$ is some constant.

After the invaded antigen has been detected, the immune response could be visualized as a negative feedback control process that maintains the concentration of the fighter cells sufficient for the complete elimination of the antigen. It could be theorized that until the intruders are present, the concentration of the fighter cells exponentially increases following the equation

$$C(t) = C(t - t_1 - \tau)e^{\delta(t-t_1-\tau)} \tag{5}$$

Note that during the time interval τ concentration C(t) changes very little, and therefore (5) could be rewritten as

$$C(t) = C(t_1)e^{\delta(t-t_1-\tau)} \tag{6}$$

where δ is a non-negative number dependent on the current concentration of the antigen as follows

$$\delta(t)\begin{cases} >0 & \text{if } P(t)>0 \\ = -\alpha & \text{if } P(t)=0 \end{cases} \qquad (7)$$

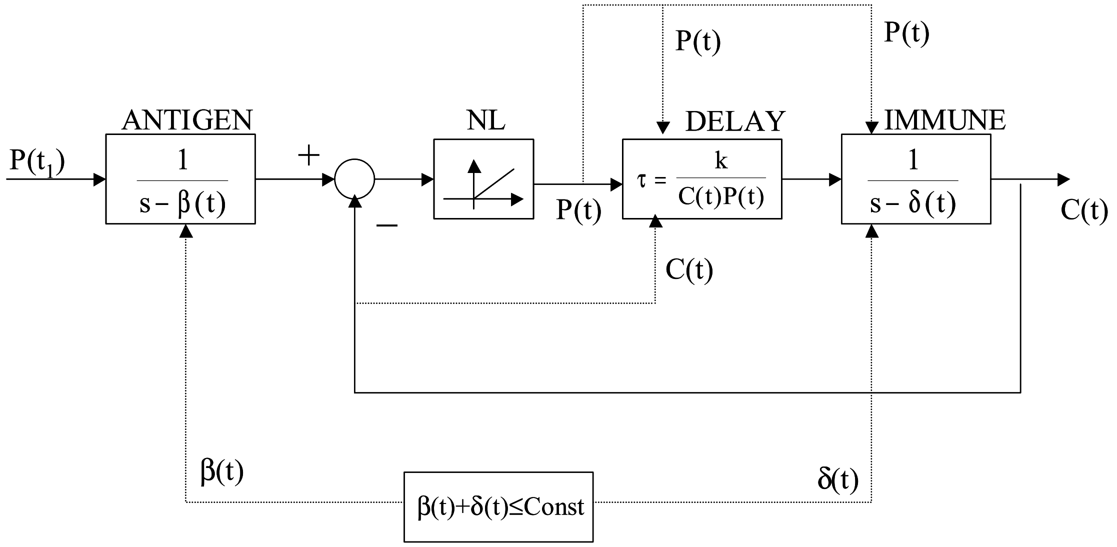
It could be seen from (7) that in the absence of antigen C(t) reverts to the natural motion pattern.

The existence and multiplication of fighter cells also consumes limited resources of the biological organism, competing for these resources with the multiplying antigen cells. The quantity $\delta(t)>0$ also could be viewed as the "available share of the resources" and therefore the following constraint should be expected:

$$K_1[\beta(t)+\delta(t)]+ K_2[C(t)+P(t)] \leq \text{const} \qquad (8)$$

where K_1 and K_2 are coefficients reflecting the "cost of multiplication" and the "cost of living" of the immune fighter cells and antigen cells.

Now we can summarize the above equations in the following block diagram representing the logistic and dynamics of the immune response and facilitating its simulation analysis. Block ANTIGEN represents the dynamics of the multiplication of the antigen cells according to equation (3). Block IMMUNE describes both the "natural" and "forced" dynamics of the multiplication of the immune fighter cells in consistence with equations (2), (6) and (7). It could be seen that the multiplication rate of both types of counteracting cells is limited as they share the limited resources of the infected biological organism. Block DELAY is responsible for the implementation of the variable delay in the immune response channel, as per equation (4). Block NL prevents the simulation system from the appearance of negative concentrations of the immune and antigen cells.



The simulation of immune responses is initiated by applying a pulse-type signal

$$P_0(t) = \begin{cases} P(t_1), & 0 \leq t \leq \Delta t \\ 0, & t > \Delta t \end{cases}$$

where Δt is the simulation time step and $P(t_1)$ represents the severity of infection. Depending on the initial choice of the simulation constants, the simulation will lead to the following outcomes:

1. Lethality: the antigen cells multiply faster than the fighter cells and consequently overwhelms the immune defenses.
2. Chronic infection: an unstable balance between the concentration of fighter cells and the concentration of the specialized immune fighter cells.
3. Full recovery: the immune fighter cells multiply faster than the antigen cells and consequently neutralize the intruder.

The following situations were subjected to simulation analysis utilizing the developed equations:

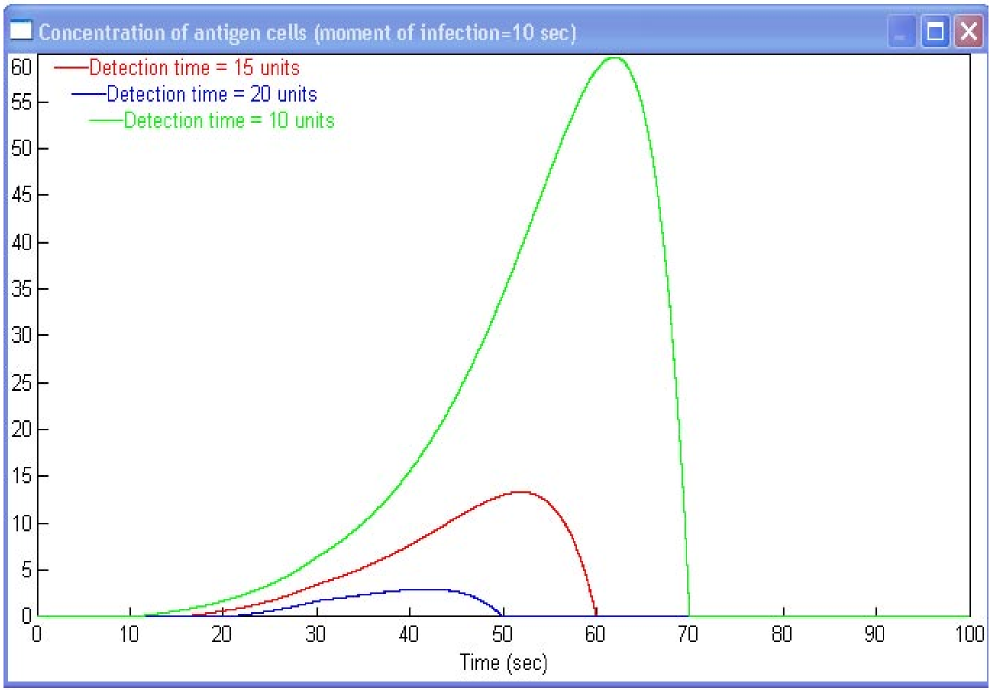
a- effect of the concentration of the antigen cells in the “full recovery” case obtained for various detection delays values. It could be seen that as the detection time increases due to low immunity to the particular antigen (low initial concentration of the specialized immune fighter cells) the severity and duration of the disease increases, see Figure below.

b - effect of the concentration of the antigen cells in the “full recovery” case obtained for various initial concentrations of the antigen cells (severity of the infection). It could be seen that due a very rapid multiplication of the antigen cells, their initial concentration manifests itself primarily by affecting the detection time.

c - effect of the concentration of the antigen cells in the “full recovery” and “lethality” cases obtained for various detection delays values. It could be seen that as the detection time increases due to low immunity, the multiplying antigen cells, before being detected and counteracted, occupy the increased share of the available recourses of the organism. Consequently, after exceeding some threshold, they prevent the immune fighter cells from successfully multiplying thus gaining the control over the organism and causing its effective death.

The described outcomes of a disease caused by infection could be easily interpreted in terms of a computer network subjected to an information attack. One can realize that the “lethality” outcome is consistent with the outcome of a successful DoS or a virus attack completely disabling a specific computer or network. The “chronic disease” case could be visualized as a partial loss of the network operation and throughput. As it was initially said, familiarity with the configuration, hardware and software components of a computer network creates the conditions when particular parameters of the equations describing the “immune response” could be established with sufficient accuracy. While some of these parameters should be viewed as constants, other parameters could be varied by reconfiguring the network, introducing novel attach detection software, “borrowing” additional resources from another network, etc. Intentional variation of the network characteristics with the purpose of achieving the assurance of the “full recovery” outcome of an information attack constitutes the control effort that could be administered by the network operator or automatically.

The consequent efforts of this research will be directed at the establishing realistic models of the network defense system dynamics, characteristics of particular types of information attacks, analysis of the “bottlenecks” of the network defense systems, and finally, development of dependable control laws.



Concentration of antigen cells (moment of infection=10 sec)
Detection time = 15 units
Detection time = 20 units
Detection time = 10 units
60
55
50
45
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100
Time (sec)

VI. Methods of Immunocomputing and Computer Network Security

Based on the principles established for the immune system, a new computational technique, called the Artificial Immune System (AIS), is rapidly emerging, which appears to offer powerful and robust information processing capabilities for solving complex problems. Like Artificial Neural Networks (e.g. see [13]), AISs can learn new information, recall previously learned information, and perform pattern recognition in a highly decentralized, efficient fashion. Researches have started using AISs in many applications including information security, vaccine design, fault detection, data mining, robotics, etc. [2].

A rigorous mathematical basis of AIS based on a biological prototype of *immune network* and on the notions of *formal protein* and *formal immune network* has been proposed in [1]. We referred this model as *formal immune system*, or *immunocomputing* (IC). This paper provides a further development of our IC approach and its application to pattern recognition. The approach reflects the fact that recognition is the basic function performed by the immune system and provides a mathematical apparatus, uniquely suitable for describing recognition as it is performed in nature, i.e. based on the features of recognizing (binding) antigen by antibodies in the immune system. In the natural immune system, local binding of immune cells and molecules to antigenic peptides is based generally on the behavior of surface proteins. In particular, immune cells contain proteins on their receptors that apparently play the key role both in immune response and recognition processes.

In this work, we consider a model of the proteins as Formal Protein (FP) coded by a real unit vector. We propose also a mathematical model of binding, or recognizing between the proteins. According to the biological prototype, the central notion of this model is binding energy. We determine the binding energy between a pair of FPs by a bilinear form over the pair of corresponding unit vectors. This bilinear form is determined by a real rectangular matrix. Thus, we obtain a convenient quantitative measure of the extent of recognition between the FPs. The low is binding energy – the better is recognition, and vice versa. Simply put, the binding energy is the energy that should be applied to bind the FPs. The positive energy corresponds to the repulsion between the FPs, while the negative energy corresponds to the attraction between them. Thus, the minimal negative energy corresponds to the strongest attraction and consequently, the best recognition between the FPs.

This model leads to the formulation of a rigorous IC approach to pattern recognition. Within the approach any pattern is considered as a set of characteristics, represented by real values. There are at least two ways to represent a pattern. First, by transforming the set into a rectangular matrix we facilitate the determination of the binding energy between any pair of FPs. This allows for establishing the pattern recognition mode by the so-called supervised learning, or training by an expert. Second, by transforming the set into the unit vector we represent the pattern as an FP. This allows for establishing the unsupervised learning, or automated classification of the patterns. In both cases some optimal set of FPs is computed for any given set of pattern samples. An optimal set of probes is used to recognize the patterns~~,~~ similar to the samples. The similarity criterion is based on the proximity between the values of the binding energy. For the supervised learning the recognition is based on the minimal value of the binding energy. More specific, the pattern is recognized by such a pair of the FPs that provides the minimum of the binding energy among all pairs of the probes. For the unsupervised learning the recognition is based on the minimal geometric distance of the binding energies between the pairs

"sample-probe" and the pairs "probe-probe". Such an approach results in rather fine classification of the patterns and provides the visualization of the classes as the points in a special plane.
As a demonstration of this IC approach to pattern recognition consider its application to the problem of intrusion detection in computer networks. In this case any pattern represents a set of the network connection records, such as duration of the connections, number of the sent and received data bytes, packages, and so on, in conjunction with normal behavior of the network and during an intrusion. The example shows how an attack could be recognized by the proposed approach.

Mathematical basis

According to [9], any unit vector P with n real-valued components

$$P = [p_1, \dots, p_n]^T, \;\; PP^T = \mathrm{I},$$

can be considered as a special kind of FP with $n-1$ links.

Binding energy between any pair of such FPs: $\{P, R\}$, of the dimensions n_P and n_R, correspondingly, can be defined by a bilinear form:

$$w = -P^T MR, \qquad (1)$$

where M is a matrix of dimension $n_P \times n_R$.

It is known (e.g. see [5]), that extreme values of the bilinear form (1) are determined by the so-called Singular Value Decomposition (SVD) of the matrix:

$$M = s_1 P_1 R_1^T + \dots + s_r P_r R_r^T, \qquad (2)$$

where s_i are singular values; P_i and R_i are left and right singular vectors; r is rank of the matrix.

Such SVD exists for any rectangular matrix over the field of real values. Singular values and singular vectors possess the following useful properties:

$$s_1 \ge s_2 \ge \dots \ge s_r \ge 0,$$
$$P_i^T P_i = 1,\; R_i^T R_i = 1,\; i = 1,\dots,r,$$
$$P_i^T P_j = 0,\; R_i^T R_j = 0,\; j \ne i.$$

Let us consider binding as recognizing between FPs. Then binding energy can be viewed as a numerical measure of the recognition: the lower this energy is – the better is recognition, and vice versa. At the same time, if matrix M is given, then the pair of the FPs with extreme recognition ability is corresponded to the first singular vectors P_1 and R_1 of this matrix, while the maximal singular value s_1 determines the minimal binding energy:

$$w^* = -P_1^T M R_1,$$
$$w^* \le w(P,R),\; \forall P,R : P^T P = R^T R = 1.$$

Such rigorous mathematical properties of the binding energy between the FPs that are based on the bilinear form (1) and the SVD (2) provides the computational basis for developing a rigorous IC approach to pattern recognition.

Pattern recognition

In general, pattern recognition can be defined as follows. Let us treat real values $x_1, \ldots, x_n$ as a set of characteristics. Consider an arbitrary vector $X=[x_1, \ldots, x_n]^T$ as a pattern that belongs to a *characteristic space* $\{X\}$. Consider, that the space can be partitioned onto the subsets (classes) $\{X\}_c$, $c=1, 2, \ldots k$. Then recognition of a sample X implies the determination of such a class c that $X \subset \{X\}_c$, while learning implies the partitioning (classification) of the characteristic space. If the space is being partitioned into known classes by experts, then it is known as *supervised learning* (or *training*). If the number of the classes k and the class definitions are unknown a priory, then the situation is referred to as *unsupervised learning*.

The use of the binding energy between the FPs as the rigorous measure of the recognition constitutes the main feature of the IC approach to pattern recognition. This approach is outlined as follows.

Supervised Learning

Folding vectors to matrices

Fold vector X of dimension $n\times 1$ to a matrix M of dimension $n_P \times n_R = n$. It has been shown formally in [7], that this operation increases the specificity of recognition.

Learning

Form matrices $M_1, \ldots, M_k$ for all classes $c=1,\ldots,k$, and compute their singular vectors:

$$\{P_1, R_1\} \text{ – for } M_1, \ldots, \{P_k, R_k\} \text{ – for } M_k.$$

Recognition

For each input pattern (*sample*) M compute k values of the binding energy between each pair of the singular vectors (*probe*):

$$w_1 = -P_1^T M R_1, \ldots, \ w_k = -P_k^T M R_k .$$

Determine the class to which the pattern belongs by the minimal value of the binding energy:

$$c : w^* = \min_c \{ w_c \} . \qquad (3)$$

Unsupervised Learning

Consider the matrix $M = [X_1 \ldots X_m]^T$ of dimension $m\times n$ formed by m vectors (patterns) $X_1, \ldots, X_m$. Consider the SVD of this matrix:

$$M = s_1 \begin{bmatrix} p_{11} \\ \ldots \\ p_{1n} \end{bmatrix} R_1^T + s_2 \begin{bmatrix} p_{21} \\ \ldots \\ p_{2n} \end{bmatrix} R_2^T + \ldots , \qquad (4)$$

where s_1, s_2 are the first two singular values, and R_1, R_2 are the right singular vectors.

Note, that every string i of the matrix M represents the values x_{ij} of n characteristics of the pattern X_i, where $i=1,\ldots,m$ and $j=1,\ldots,n$. Hence, according to the SVD properties, the components p_{1i}, p_{2i} of the left singular vectors P_1, P_2 satisfy the following equations:

$$p_{1i} = P(X_i^T)\frac{|X_i|}{s_1}R_1,\; p_{2i} = P(X_i^T)\frac{|X_i|}{s_2}R_2, \qquad (5)$$

where

$$|X| = \sqrt{x_1^2 + \ldots + x_n^2},\; P(X) = \frac{X}{|X|}.$$

Comparison of (1) and (5) makes obvious, that the components p_{1i}, p_{2i} can be computed as binding energies w_{1i}, w_{2i} between the FP $P(X_i)$ and the FPs R_1, R_2, correspondingly. Thus, every vector X_i with $n \geq 2$ characteristics can be mapped onto only two values of binding energies.

Such mapping gives a mathematically rigorous way to represent and view all patterns, regardless of the number of characteristics, as points in two-dimensional space of binding energies $\{w_1, w_2\}$. This plane could be treated also as a *shape space* of the IC, according to [3]. Such representation of patterns in the shape space of the IC allows classifying them in a natural way by the groups (*clusters*) of the neighboring points. Such classifying can be performed by experts using supervised training as well as without experts using unsupervised learning.

Therefore, the technique of the unsupervised learning by the IC approach is as follows.

First, the matrix $M= [X_1\ldots X_m]^T$ is formed by all samples $X_1, \ldots , X_m$.

Second, the SVD of this matrix (2) is computed to obtain the first r singular values $s_1,\ldots,s_r$ and singular vectors $P_1, R_1 ,\ldots, P_r, R_r$.

Third, two right singular vectors R_{m1}, R_{m2} are chosen as the "antibodies", or the probes. Usually, the first two singular vectors R_1, R_2 are to be chosen. However, in some cases another pair of the probes can result in even more fine classification of the patterns.

Forth, all initial samples $X_1, \ldots , X_m$ are mapped onto the plane, according to (5).

Fifth, the clusters (classes) of the patterns are formed, according to the proximity of the corresponding points in the plane. A vector norm can serve as the measure of the proximity, any method used in the traditional pattern recognition (e.g. see review in [7]) can be employed as the method of forming clusters.

Sixth, the recognition of any sample X can be performed by mapping it onto the plane as the point, according to (5). Then the class of the sample is determined by the cluster of the nearest points in the plane.

The IC approach to pattern recognition has already appeared to be useful in solving a number of important practical tasks, including detection of dangerous situations in near-Earth space [9], detection of the similarity in the dynamics of the infectious diseases in Russia [7], computation of the map of complex appraisal of environmental conditions in the city of Kaliningrad [6], and the utilization of space-time dynamics of the plague infection in Central Asia for health risk assessment [11]. Consider now a possible application of the IC approach to intrusion detection in computer networks.

Application to intrusion detection

A numerical experiment was staged using a fragment of the database by [1] representing several types of intrusions, as shown in Tab. 1 and Tab. 2.

First column of Tab. 1 presents conventional names of the 15 intrusion types (apache2,…, xsnoop) and a normal behavior of the network connection (normal). Second column assigns short names to the intrusion types as listed in Tab. 2. The rest columns of Tab. 1 show auxiliary conditions under which the data of the intrusions had been recorded (columns "1",…,"33" in Tab. 2).

Table 1. Types of intrusions

Intrusion type	Sign	Protocol type	Service
apache2	ap2	Tcp	http
buffer_overflow	b_o	Tcp	telnet
guess_passwd	g_p	Tcp	pop_3
ipsweep	ips	Icmp	eco_i
multihop	mul	Tcp	telnet
named	nam	Tcp	domain
normal	norm	Udp	private
phf	phf	Tcp	http
pod	pod	Icmp	ecr_i
portsweep	por	Tcp	private
saint	st	Tcp	private
sendmail	se	Tcp	smtp
snmpgetattack	snm	Udp	private
udpstorm	ud	Udp	private
xlock	xlo	Tcp	X11
xsnoop	xsn	Tcp	X11

The database fragment in Tab. 2 below presents 106 records (column "#") on the several types of the intrusions (column "Sign"). The fragment utilizes 33 characteristics (columns "1",…,"33") for the network connection records, including lengths (number of seconds) of the connection (column "1"), number of data bytes from source to destination (column "2"), from destination to source (column "3"), and so forth (the rest of columns).

These data were represented in the shape space of the IC as follows. A matrix M of dimension 106×33 was formed by all input data of Tab. 2 (rows "1" – "106" and columns "1" – "33"). The SVD of this matrix was computed according to (4). For guaranteed classification of intrusions this SVD involved 6 singular values $s_1, \dots , s_6$ and corresponded pairs of left and right singular vectors $P_1, R_1, \dots , P_6, R_6$. The last two right singular vectors R_5, R_6 were taken as two "antibodies" {FP-1, FP-2}. Accordingly, every string M_i (row of Tab. 2) of the matrix M, where $i=1,...106$, was considered as one of the following FPs: {FP$_1$, ... , FP$_{106}$}. Then, according to (1) and (5), the components of the left singular vectors P_5 and P_6 represented two values $\{w_1 , w_2\}$ of the binding energy between any FP$_i$ and each "antibody":

$$w_{1i} = w(\text{FP-1}, \text{FP}_i), \quad w_{2i} = w(\text{FP-2}, \text{FP}_i),$$

Table 2. Intrusion records (columns "1"..."33") and shape space coordinates of the intrusions (columns "w_1", "w_2")

#	Sign	1	2	...	33	w_1	w_2
1	ap2	906	57964		0	-0.044	-0.018
...	...	...	...	...	...	...	...
11	b_o	198	2442		0	0.117	-0.082
...	...	...	...	...	...	...	...
26	mul	69	331		0	0.116	-0.088
...	...	...	...	...	...	...	...
45	norm	0	223		0	-0.216	0.065
...	...	...	...	...	...	...	...
51	phf	0	0		0.06	-0.008	0.003
52	pod	0	1480		0	0.113	-0.090
...	...	...	...	...	...	...	...
62	por	0	0		1	0.113	-0.081
...	...	...	...	...	...	...	...
79	st	0	0		0	0.119	0.361
...	...	...	...	...	...	...	...
82	se	2	4485		0	0.112	-0.089
...	...	...	...	...	...	...	...
88	snm	0	105		0	-0.132	-0.014
...	...	...	...	...	...	...	...
98	ud	0	0		0	0.001	0.004
99	xlo	199	56124		0	0.108	-0.077
...	...	...	...	...	...	...	...
106	xsn	50	226		0	0.114	-0.088

All these values are shown also in Tab. 2 (columns "w_1" and "w_2") and represented geometrically in Fig. 1.

Therefore, any intrusion record with 33 characteristics (see rows of Tab. 2) is represented in Fig. 1 as a point in the IC shape space. A number of the point corresponds to the number of the intrusion record from Tab. 2. Horizontal coordinate of the point corresponds to the value w_1 and vertical coordinate corresponds to the value w_2.

As one can see from Fig. 1 below, the main advantage of such a representation is that the points form clear groups (classes) in the shape space. Such classes (rounded in Fig. 1 for the sake of obviousness) correspond to the normal behavior and the intrusion types. In other words, we have obtained a classification of the intrusions by the unsupervised learning. The obtained classification shows that the IC approach separates surely normal behavior from intrusions. Indeed, the points, which correspond to the normal behavior ("norm" points ## 45, 46, 49, 50, as well as the point #48, can be separated clearly from the other points, which correspond to intrusions. Moreover, Tab. 2 and Fig. 1 allow for reliable detection (separation) almost all types of the intrusions as relatively close points in the shape space. Zooming in with Fig. 1 can produce more thorough analysis within the types of intrusions.

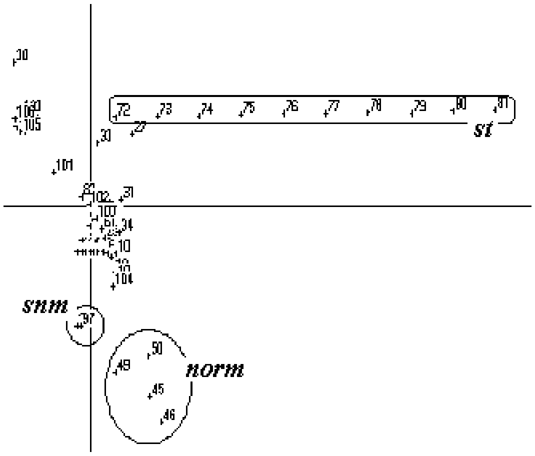


Figure 1. Intrusion records in the IC shape space.

Using such classification, the IC approach could recognize the intrusion in the following fashion.

Any network connection data should be recorded as a vector with 33 components:

$$X = [x_1, \ldots , x_{33}]^T .$$

The values of the components of this vector correspond to the characteristics "1" – "33" of Tab. 2. Then two values w_1, w_2 of the binding energy should be computed according to the bilinear forms in (1) and (5):

$$w_1 = \frac{1}{s_5} X^T R_5 , \quad w_2 = \frac{1}{s_6} X^T R_6 .$$

The point-sample with such coordinates could be compared with any point-probe in the shape space to find the closest one. This proximity could be quantified by Euclidean norm or any other norm. Thus, the closest point-probe could determine either a normal behavior, or a type of intrusion for the point-sample, which represent the recognizing type of the network connection.

For example, consider the row #45 of Tab. 2 as a set of characteristics of a checking network connection. Then, according to Fig. 1, the closest point-probe in the shape space is a point #46. As the point-probe #46 corresponds to the normal behavior, then the point-sample #45 could be recognized as the normal behavior of the checking network connection.

Another method of intrusion detection by the IC approach could be also based on the supervised learning (see section III.A) without using shape space representation. In this case any network connection record with 33 components should be folded to a matrix M of dimension 3×11 as follows:

$$M = \begin{bmatrix} x_1 & \ldots & x_{11} \\ x_{12} & \ldots & x_{22} \\ x_{23} & \ldots & x_{33} \end{bmatrix} .$$

In the learning mode a training set of records should be used to form matrices M_{app2} , ... , M_{xsn} for the types of intrusions and a matrix M_{norm} for the normal behavior. SVD of each matrix results in a pair of the fist left and right singular vectors $\{P_1, R_1\}$ serving as a pair of probes ("antibodies") for each learning class:

$$\{P_1, R_1\}_{app2} , \ \dots , \ \{P_1, R_1\}_{xsn} , \ \{P_1, R_1\}_{norm} .$$

In the recognition mode, any checking record X should be folded to the matrix M. Then 16 values of the binding energy between the probes should be computed:

$$w_{app2} = -\{ P_1^T \}_{app2} M \{ R_1 \}_{app2} \ , \ \dots \ ,$$
$$w_{xsn} = -\{ P_1^T \}_{xsn} M \{ R_1 \}_{xsn} \ ,$$
$$w_{norm} = -\{ P_1^T \}_{norm} M \{ R_1 \}_{norm} \ .$$

According to (3), intrusion type could be recognized by the minimal value of the binding energy:

$$w^* = min\{ w_{app2}, \dots, w_{xsn}, w_{norm} \} .$$

Simply put, the class of the pair of probes that corresponds to the minimal value of the binding energy is the type of intrusion or the normal behavior.

To improve intrusion detection, the two methods of recognition, supervised learning and unsupervised learning using shape space could be combined.

Intrusion features analysis

The proposed approach is able also to reduce essentially the number of recording characteristics of the network connection for intrusion detection. In other words, the approach allows selecting most useful features of the network traffic to distinguish normal connection from attack. In could be done as follows.

According to the previous section, consider the coordinates R_5, R_6 of two "antibodies"-probes shown in Tab. 3. Every pair of these coordinates (row of Tab. 3) represents one of the 33 characteristics (first column of Tab. 3) of the network connections used in the fragment of the database of our numerical experiment.

Representation of these coordinates as points in the shape space of the IC allows to select the most distinctive points that stand somehow out relatively to a background of other points. Such distinctive points are picked out as bolded rows ## 1, 15, 16, 24, 25, 26, 33 in Tab. 3 below. As one can see, these points have relatively big values of their coordinates.

Another numerical experiment was staged using this reduced set of seven characteristics. According to the previous section, a matrix M of dimension 106×7 was formed by input data of Tab. 2 (rows "1" – "106" and columns "1", "15", "16", "24", "25", "26", "33"). The SVD of this matrix was computed according to (4), and so forth. Finally, the shape space representation of the data was computed. Remarkably, that the reduced set of 7 selected characteristics gives almost the same results, as the full set of 33 characteristics in Fig. 1. The reduced set also separates surely normal behavior from intrusions and detects reliably almost all types of the intrusions as relatively close points in the shape space.

Table 3 . Coordinates of "antibodies"-probes

characteristic #	R_5	R_6
1	**0.0883**	**0.0002**
2	0.0000	-0.0000
...	...	...
14	0.0001	-0.0004
15	**0.0856**	**0.9950**
16	**-0.0104**	**0.0157**
17	0.0003	0.0015
...	...	...
23	-0.0001	0.0002
24	**0.4163**	**-0.0777**
25	**-0.9008**	**0.0585**
26	**-0.0037**	**0.0013**
27	0.0011	0.0008
...	...	...
32	0.0007	0.0014
33	**0.0023**	**0.0052**

In other words, the IC approach has reduced initial data of network connection records almost to five times from 33 characteristics to 7 without loosing intrusion detection possibilities. Therefore, the approach is able to select most useful traffic features to distinguish normal connection from attack. The description of these features for the data of the computation experiments is given in Tab. 4.

Table 4. Most useful traffic features to detect attack

feature #	feature name	description
1	Duration	length (number of seconds) of the connection
15	Count	number of connections to the same host as the current connection in the past two seconds
16	srv_count	number of connections to the same service as the current connection in the past two seconds
24	dst_host_ count	number of connections to the same destination host as the current connection in the past two seconds
25	dst_host_srv_ count	number of connections to the same destination host and the same service as the

		current connection in the past two seconds
26	dst_host_ same_ srv_rate	% of connections to the same destination host and the same service as the current connection in the past two seconds
33	dst_host_srv_ rerror_rate	% of connections that have errors

Conclusion

As shown also in [11], the proposed IC approach to pattern recognition is rather powerful, robust and flexible. It is able to give rather fine classification thus focusing attention on the most dangerous situations, which is beyond the possibilities of the traditional statistics.

We would also like to highlight three features, which make the approach promising:

1. Appropriate biological prototype of pattern recognition by the immune system proteins honed by million-year evolution;
2. Rigorous mathematical basis of the IC;
3. Possibility of hardware implementation of the IC in a special immunochip [1].

Such an implementation could significally enhance the level of reliability, flexibility and computational efficiency of AISs as well as their principal applications (e.g. to information security, according to [2], [12]) to a new level of maturity. On the other hand, a growing need to overcome main disadvantages of the neural networks' models, such as spurious patterns, small storing capacities comparatively to the dimension of networks, non-localized errors, etc., is well recognized. These factors prevent the proliferation of neurochips in those fields where the cost of a single error is too high (e.g. aviation, medicine, information security). It could be said that the natural immune system successfully protects organism from such "errors" and invaders. Therefore, the authors hope that the IC has the potential for serving as immune systems in large-scale control and computer networks.

VII. Mining of the Data Traffic in a Computer Network and Detection of DoS Attacks

1. Denial of Service attacks on computer networks

The vulnerability of modern computer networks to information attacks is of great concern to any organization utilizing computer networks. With the increase of size, interconnectivity and number of users, networks are becoming increasingly vulnerable to newly developed direct and remote threats and attacks compromising the integrity, confidentiality or availability of a network, and consequently information.

Denial of Service (DoS) attacks making crucial network resource and/or information unavailable, are among the most common attack types. DoS attacks are both effective and easy to deploy. DoS attacks threaten all systems connected to the Internet, including servers, clients, routers, and firewalls. In a DoS attack scenario, the attacker sends malicious traffic to the target with the aim of crashing, crippling, or jamming communication between the target system and legitimate users. Current computer security systems provide some degree of protection of information attacks and enhance the decision-making ability of the network administrator. These functions are performed by a number of independent system components requiring an enormous amount of distributed and specialized knowledge, and consequently, computations. As a rule, these systems represent a bottleneck with regard to throughput, speed, reliability and flexibility of a network

Unlike hardware failures of network components, information attacks unravel in time, propagating through the network and affecting the increasing number of users and hosts. A network operator could detect these undesirable developments at relatively early stages, and undertake important decisions alleviating the attack. These decisions typically reflect experience and intuition of human operator and consequently are based on his/her subjective interpretation of the current status of the network. The dynamic nature of information attacks and ability to monitor status of a network facilitate the development of mathematically justified, more efficient approaches to the detection of attacks providing important insight to human operators. This research is aimed at the application of advanced statistical analysis for the objective assessment of the network status and detection of unique changes in the status indicative of information attacks. Implementation of this approach can result in an information security tool facilitating early detection of the DoS-type attacks and providing on-line support of operators' decisions.

2. Quantitative characterization of a computer network

A modern computer network constitutes a complex dynamic system containing many relatively independent, highly interconnected, dynamic components. While the structural complexity of such a network is not to be questioned, its dynamic nature is not that obvious. Indeed, electronics-based network hardware has negligible dynamics. However, every data processing, digital communication, and data interpretation procedure, implemented in software, has finite execution time. Within the network, operation of every logical unit can be characterized by the data traffic flow through this unit that comprises data packets of various sizes. The composition of data flow could be quantified by percentages of large, medium and small size packets. The packets may be accumulated in queues and are processed in some order that results in the effect of "inertia" responsible for the network dynamics. Unlike classical view of system dynamics, dynamics of a computer network is both system- and input-dependent.

System dependence of network dynamics is related to the throughput of particular modules of the network. Input dependence manifests itself by time-varying number of packets to be processed and their composition, and specific computational tasks.

It is understood that while the network configuration and throughput of its components are not affected by information attacks, they do not carry any useful information for our purpose. However, information attacks cause gradual changes in the volume and composition of particular data flows within the network, ultimately leading to a complete denial of service. While periodic changes of the flows volumes and compositions could be observed under normal operation of the network on the 24-hour and 7-day scales, it is expected that changes in the flow volumes and compositions caused by information attacks follow very different statistical patterns. This creates the opportunity for differentiating "normal" fluctuations from those caused by attacks. These realities have led the authors to the following conclusion: *dynamic properties of a computer network, representing its immediate status, could be best described in a specially defined state space that reflects not the configuration of the network hardware or software, but the volumes and compositions of data flows through particular modules of the network, and their rates of change.*

Assume that a computer network consists of M interrelated processing modules, and the m-th module has the data flow characterized by

- volume, i.e. number of bytes per second, - z_m,

- composition of the data flow, i.e. the percentage of large, medium and small packets – $\mathrm{p_m}$, r_m, s_m, such that $p_m+r_m+s_m=1$, and

- time derivatives (rates of change) of these variables – z'_m , p'_m , r'_m , s'_m , where $m=1,2,\ldots,M$.

For convenience, all "state variables" of the computer network could be assembled under the state vector

$$X = [z_m , p_m , r_m , s_m , z'_m , p'_m , r'_m , s'_m ,\ m=1,2,\ldots,M] \qquad (2.1)$$

It could be seen that the attempt to quantify the status of a computer network leads to the definition of the state space of high dimension. It is also understood that X is a random vector and its realization $X(k)$ represents the an immediate status of the network at the discrete time moment $k=1,2,3\ldots$. The availability of a network monitoring system allows us to accumulate a database, $\{X(k)\ , k=1,2,3\ldots, N\}$, representing the network status (operation) over some period of time. Although the established state space and database may not be sufficient for the formulation of the network control problems, it can provide sufficient information for the detection of specific changes in the network status caused by information attacks. Fortunately, attack detection/recognition is a much simpler task than control that may concentrate on the analysis of some "bottlenecks" of the network. Application of statistical analyses facilitates the detection of the "informative" components of the state vector thus resulting in the significant simplification of the attack detection/prediction problem, and consequently, definition of simplistic recognition rules. It could be achieved if, and only if, the available database contains both the normal operational data, and the information accumulated during attack of the known type, i.e. has the format $\{X(k), Q(k), k=1,2,3\ldots, N\}$, where $Q(k)$ is a flag indicating the presence of the attack at the discrete time moment k.

Recall that the particular realizations of the database are random vectors. Application of a statistical approach to the detection/prediction problem results in "averaging out" the effects of such unimportant factors and fluctuations of variables *X* during normal operation of the network, and detecting abnormal trends in the network status data, that could be interpreted as an incipient information attack. The following mathematical techniques are consistent with the statistical nature of the attack detection/prediction problem.

3. Cluster Analysis and Genetic Optimization

Cluster analysis is a group of statistical techniques facilitating the detection of informative components of what could be a very extensive database. It is clear that this task cannot be accomplished without relevance to some decision-making or a classification problem. We will visualize the database as a combination of realizations of real status variables, X, and a binary class indicator, Q:

$$\{X(k), Q(k)\} = \{x_1(k), x_2(k), x_3(k), \ldots, x_n(k), Q(k)\} \tag{3.1}$$

where $k = 1,2,3,\ldots,N$ is the realization index, and $Q(k)$ can have only two alternative values, "**a**" or "**b**". Then the classification rule is established on the basis of some function defined in the X space, $F[X]$, such that, generally, $F[X] \le 0$ for the majority of realizations when $Q(k) = \text{"a"}$ and $F[X] \succ 0$ for the majority of realizations when $Q(k) = \text{"b"}$, or in terms of conditional probabilities,

$$P\{F[k] \le 0 \;/\; Q[k] = \text{"a"}\} \succ P\{F[k] \le 0 \;/\; Q[k] = \text{"b"}\} \tag{3.2}$$

where $P\{A / B\}$ denotes the probability of event *A* subject to the occurrence of event *B*.
It is also understood that the classification problem does not have a unique solution, and there is a wide class of functions $F[X]$ that could satisfy condition (3.2) to a greater or a lesser extent. A simplification of the classification rule requires reducing the number of the components of vector X to the necessary minimum by choosing the smallest group of informative components that, in combination, allow for achieving reliable classification.
Selection of the informative components implies that contributions of particular groups of components of vector X to the classification are to be evaluated, and the most contributive group(s) be chosen for the definition of the classification rule. One can realize that in order to achieve the required discrimination power of the selection procedure, the groups must be small, and in order to consider combined effects of several variables must include at least two variables [1, 2]. Consider all possible combinations of two variables taken out of *n*, where *n* is the dimension of vector X. It could be said now that the classification problem, originally defined in the space X, now will be considered on particular two-dimensional subspaces, $x_i \bigcap x_j$, where $i, j = 1,2,\ldots,n$, and $i \ne j$.

Assume that the entire array of points, marked as "**a**" or "**b**", defined in the *n*-dimensional space X by the database (3.1), is projected on particular two-dimensional subspaces (planes). Let us visualize possible distributions of these projections. Figure 1 below illustrates a subspace that has no potential for the development of a classification rule due to the fact that points marked as "**a**" and "**b**" are distributed quite uniformly in this plane. The subspace of Figure 2 indicates a certain degree of separation between points "**a**" and "**b**" and, therefore, should be viewed as informative. Figures 4, 5, 6 also illustrate possible cases of separation pattern in informative subspaces.

As shown in Figures 4, 5, 6, a correlation ellipse, properly defined in the particular informative subspace, presents an ideal choice of the separating function. Figure 3 indicates that size, shape, position, and orientation of such an ellipse are defined by five parameters: coordinates of two focal points, $\left[\alpha_1, \beta_1\right]$, $\left[\alpha_2, \beta_2\right]$ and the constant δ, such that for any points of the ellipse, $\left[x_i, x_j\right]$, the following equation holds,

$$\sqrt{\left(x_i - \alpha_1\right)^2 + \left(x_j - \beta_1\right)^2} + \sqrt{\left(x_i - \alpha_2\right)^2 + \left(x_j - \beta_2\right)^2} = \delta \tag{3.3}$$

Similarly, equations

$$\sqrt{\left(x_i - \alpha_1\right)^2 + \left(x_j - \beta_1\right)^2} + \sqrt{\left(x_i - \alpha_2\right)^2 + \left(x_j - \beta_2\right)^2} \le \delta \tag{3.3a}$$

$$\sqrt{\left(x_i - \alpha_1\right)^2 + \left(x_j - \beta_1\right)^2} + \sqrt{\left(x_i - \alpha_2\right)^2 + \left(x_j - \beta_2\right)^2} \succ \delta \tag{3.3b}$$

represent any point $\left[x_i, x_j\right]$ within and outside the ellipse.

Consider the problem of the optimal definition of parameters $\left[\alpha_1, \beta_1, \alpha_2, \beta_2, \delta\right]$ of a correlation ellipse for a particular separation pattern in the plane comprising variables x_i and x_j. According to condition (3.2), this problem could be interpreted as the minimization of a loss function that includes a "penalty" for any point "a" outside the ellipse,

$$R^a(k) = R^a\left[x_i^a(k), x_j^a(k)\right],$$

and a "penalty" for any point "b" within the ellipse,

$$R^b(k) = R^b\left[x_i^b(k), x_j^b(k)\right], \text{ i.e.}$$

$$L(\alpha_1, \beta_1, \alpha_2, \beta_2, \delta) = \sum_{k=1}^{N^a} R^a(k) + \sum_{k=1}^{N^b} R^b(k) \tag{3.4}$$

where N^a and N^b are number of points "a" and "b" in the database. Intuitively, these penalties are defined as follows:

$$R^a(k) = \begin{cases} 0, \text{ if point } \left[x_i^a(k), x_j^a(k)\right] \text{ satisfies condition (3.3a)} \\ \dfrac{1}{\left[x_i^a(k) - \mu_i^a\right]^2 + \left[x_j^a(k) - \mu_j^a\right]^2}, \text{ if point } \left[x_i^a(k), x_j^a(k)\right] \text{ satisfies condition (3.3b)} \end{cases}$$

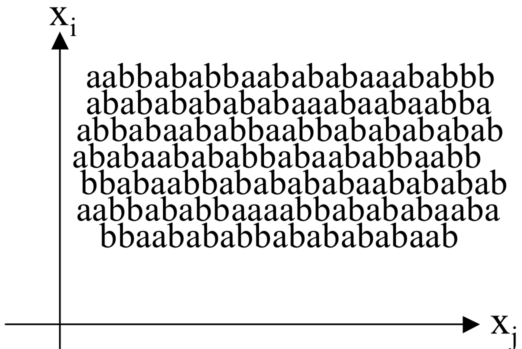

x_i
aabbababbaababababaaababbb
ababababababaaabaabaabba
abbabaababbaabbababababab
ababaabababbabaababbaabb
bbabaabbababababaabababab
aabbababbaaaabbababababaaba
bbaabababbababababaab
x_j

Figure 1

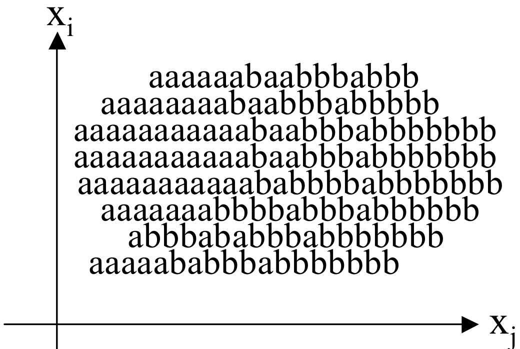


Figure 2

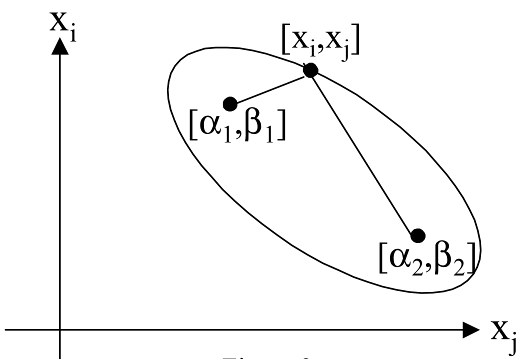


Figure 3

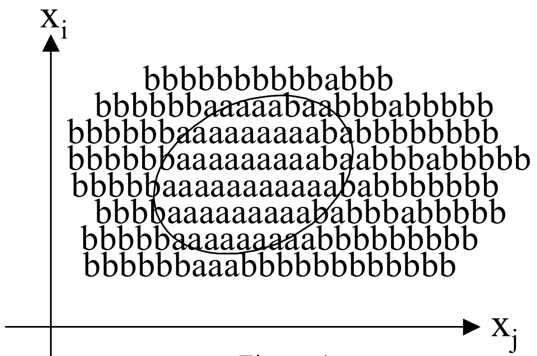

x_i
bbbbbbbbbbabbb
bbbbbbaaaaabaabbbabbbbb
bbbbbbaaaaaaaaababbbbbbbb
bbbbbbaaaaaaaaabaabbbabbbbb
bbbbbaaaaaaaaaaababbbbbbb
bbbbaaaaaaaaaababbbabbbbb
bbbbbaaaaaaaaabbbbbbbbb
bbbbbbaaabbbbbbbbbbbb
x_j

Figure 4

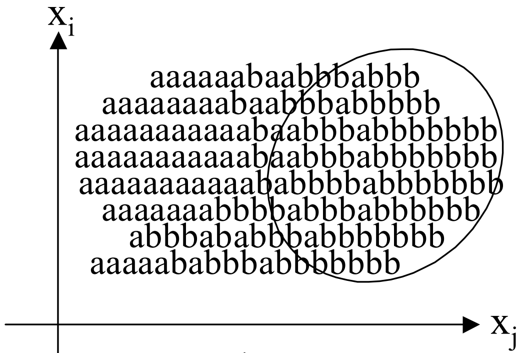


Figure 5

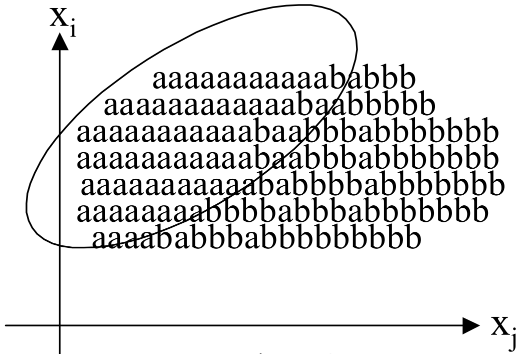


Figure 6

and

$$R^b(k) = \begin{cases} 0, \text{ if point } \left[x_i^b(k), x_j^b(k)\right] \text{ satisfies condition (3.3b)} \\ \dfrac{1}{\left[x_i^b(k) - \mu_i^b\right]^2 + \left[x_j^b(k) - \mu_j^b\right]^2}, \text{ if point } \left[x_i^b(k), x_j^b(k)\right] \text{ satisfies condition (3.3a)} \end{cases}$$

where $\left[\mu_i^a, \mu_j^a\right]$ and $\left[\mu_i^b, \mu_j^b\right]$ are coordinates of the geometric centers of points "**a**" and points "**b**" distributed in the plain $x_i \bigcap x_j$. Such a choice of penalty functions places highest emphasis on the points in the immediate vicinity of geometric centers.

It could be seen that the loss function (3.4) is not only nonlinear but also discontinuous with respect to the unknown parameters of the separation ellipse $\left[\alpha_1, \beta_1, \alpha_2, \beta_2, \delta\right]$. Therefore our attempt to obtain the numerical values of these parameters by minimizing this loss function leads to a highly nonlinear multivariable optimization problem that does not have an analytical solution. Moreover, finding its global solution numerically would also be a very difficult task. Such an optimization problem presents an ideal application for a genetic optimization procedure that combines the advantages of both direct and random search [3, 4]. Application of a genetic algorithm results in the definition of an ellipse that indeed contains the largest possible number of points "**a**", N^{aa}, and the smallest possible number of points "**b**", N^{ab}. Then the "goodness" of the ellipse-based separating rule could be characterized by the following two quantities:

$$P_{in}\{a/a\} \approx \frac{N^{aa}}{N^a} \text{ and } P_{in}\{a/b\} \approx \frac{N^{ab}}{N^b} \tag{3.5}$$

representing the probabilities of a point "**a**" and a point "**b**" to be found *within* the ellipse, see Figure 4.

Should we assume that the obtained classification rule, reflecting some compromise solution, could not be further improved? In our experience an alternative classification rule could be obtained by establishing an ellipse containing as many points "**b**", N^{bb}, and as few points "**a**", N^{ba}, as possible. This task is accomplished by the appropriate modification of the penalty functions. The resultant separating rule is characterized by:

$$P_{out}\{a/a\} \approx 1 - \frac{N^{ba}}{N^a} \text{ and } P_{out}\{a/b\} \approx 1 - \frac{N^{bb}}{N^b} \tag{3.6}$$

representing the probabilities of a point "**a**" and a point "**b**" to be found *outside* the ellipse, see Figure 5. The final selection of a separating rule should implies the comparison of ratios

$$\rho_{in} = \frac{P_{in}\{a/a\}}{P_{in}\{a/b\}} \text{ and } \rho_{out} = \frac{P_{out}\{a/a\}}{P_{out}\{a/b\}} \tag{3.7}$$

obtained for the "inside the ellipse" and "outside the ellipse" rules, and choosing the rule that results in the largest ρ value.

While the application of the genetic optimization results in the optimal characteristics of both ellipses, the final choice of ratio (3.7) provides an explicit informativity measure of the subspace $x_i \bigcap x_j$ where the ellipses are established thus facilitating the selection of the most informative subspaces. Unfortunately, genetic algorithms emulating evolutional developments in nature are too slow for being applied to the entire multitude of possible combination of two components out of n, especially if n>20. Therefore, this task should be performed on the basis of a less rigorous informativity measure, the weighted average distance between points "**a**" and "**b**" in the particular subspaces,

$$\rho_{ij} = \frac{1}{N^a N^b \sigma_i \sigma_j} \sum_{k=1}^{N^a} \sum_{m=1}^{N^b} \sqrt{\left[x_i{}^a(k) - x_i{}^b(m)\right]^2 + \left[x_j{}^a(k) - x_j{}^b(m)\right]^2} \qquad (3.8)$$

where

σ_i are σ_j are standard deviations of variables x_i and x_j.

Finally, the proposed procedure implies

1. Definition of all two-dimensional subspaces of the space X
2. Computation of the informativity measure (3.8) for every subspace $x_i \bigcap x_j$
3. Selection of a number M of the most informative subspaces
4. Selecting one of the M informative subspaces
5. Establishing the "inside the ellipse" classification rule by the application of a genetic optimization procedure, and computation of the ρ_{in} value for this subspace
6. Establishing the "outside the ellipse" classification rule by the application of a genetic optimization procedure, and computation of the ρ_{out} value for this subspace
7. Selection of the classification rule that has the largest ρ value for this subspace, and return to Step 4, until the list of informative subspaces will be exhausted.

Proliferation of genetic optimization algorithms, possessing the advantages of known random and direct search optimization procedures, combined with the availability of high performance computers alleviates major obstacles in the way of the solution of multivariable nonlinear optimization problems. The following figure illustrates application of a genetic algorithm to the solution of a classification problem mentioned above.

The algorithm proceeds as follows. Combinations of an ellipse parameters represented by vector $P=\left[\alpha_1, \beta_1, \alpha_2, \beta_2, \delta\right]$ form 5-dimensional space S. Since those parameters have bounded values, consistent with the initial variables X of the classification problem, an acceptable solution will be within a subspace, whose boundaries are defined by the following inequality $P_1 \le P \le P_2$. The algorithm forms an initial grid within this subspace by generating K uniformly distributed points P_i, $(i=1,2,\ldots,K)$, that represent possible solutions of the optimization problem.

Each point P_i of the initial grid represents an ellipse that is being assessed for the classification rule by computing the corresponding loss function $L(P_i)$ defined in (3.4), where

$$P_i = (\alpha_{1i}, \beta_{1i}, \alpha_{2i}, \beta_{2i}, \delta_i).$$

After the loss function $L(P_i)$ has been calculated at each point of the initial grid, the first generation of ellipses is formed by selecting $N_g<K$ points P_j, $(j=1,2,\ldots,N_g)$ with the smallest values of $L(P)$.

The process of parenting involves producing N_o offsprings per each of the $(N_g-1)*N_g/2$ possible pairs of points from the initial generation. Each offspring is a new point in the 5-dimensional space located on the line connecting its parents P_i and P_j $(i\neq j)$, and selected in a random fashion. At the end of this process the total population becomes $N_o*(N_g-1)*N_g/2+N_g$ points.

Parenting is followed by the mutation stage. Each point from the expanded population produces N_m mutations (points generated randomly in the immediate area). Thus, the total population count grows to

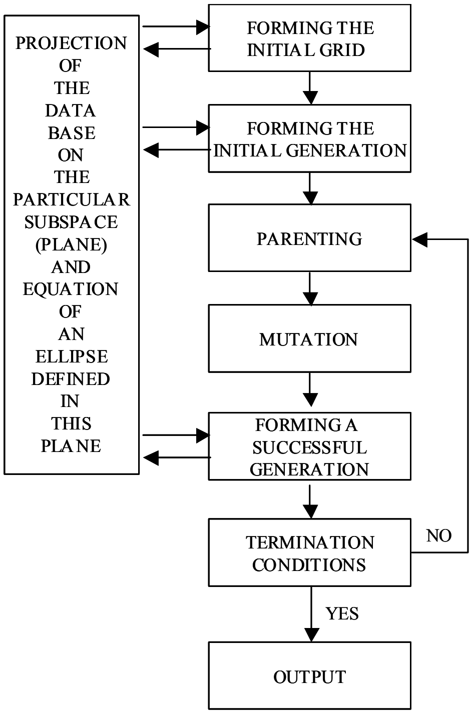

PROJECTION OF THE DATA BASE ON THE PARTICULAR SUBSPACE (PLANE) AND EQUATION OF AN ELLIPSE DEFINED IN THIS PLANE
FORMING THE INITIAL GRID
FORMING THE INITIAL GENERATION
PARENTING
MUTATION
FORMING A SUCCESSFUL GENERATION
TERMINATION CONDITIONS
NO
YES
OUTPUT

Figure 7. Genetic optimization for optimal ellipse definition

$(N_m+1)(N_o*(N_g-1)*N_g/2+N_g)$ points. A successive generation is formed by computing the loss function for all the points and selecting N_g points with the smallest values of the loss function $L(P)$.

The processes of parenting, mutation and forming a successive generation are repeated until a termination condition is satisfied. The optimization routine produces the output in the form of a vector

$$P^{OPT}=(\alpha_1^{OPT},\ \beta_1^{OPT},\ \alpha_2^{OPT},\ \beta_2^{OPT},\ \delta^{OPT})$$

which represents an ellipse resulting in the best separation rule attainable for the particular combination of two variables of the initial classification problem.

4. Bayesian Decision Making

Statistical analysis of the network data, representing normal operation of the network and its operation under known DoS attacks, results in the extraction and formalized representation of knowledge of various effects of the attack on the network. This information is attack- , host- and network-specific, and can be used for early attack detection, and potentially, for the type of attack recognition. The following mathematical framework is suggested for the attack detection scheme.

Assume that the preliminary cluster analysis utilizing the informativity criterion (3.8) has resulted in the set of M two-dimensional informative subspaces. Then the set of M respective, either "inside the ellipse" or "outside the ellipse" classification rules, $R_i[X(k)],\ i=1,2,3,..M$, was developed by the application of genetic optimization. One can realize that each rule utilizes only those two components of vector X that constitute the i-th informative subspace. For simplicity, assume that each classification rule is designed to return a negative value for the majority of points $X(k)$ representing an attack situation (and marked by an "a"). It is expected that every vector X^a, representing an attack, and every vector X^b, representing a normal network operation, would satisfy only some classification rules but not all of them. Consider the following random events:

$$
\begin{aligned}
&E_1\!: \ R_1[X(k)]\le 0 \cap R_2[X(k)]\le 0 \cap R_3[X(k)]\le 0 \cap R_4[X(k)]\le 0 \cap \ldots\ R_M[X(k)]\le 0\\
&E_2\!: \ R_1[X(k)]> 0 \cap R_2[X(k)]\le 0 \cap R_3[X(k)]< 0 \cap R_4[X(k)]\le 0 \cap \ldots\ R_M[X(k)]\le 0\\
&E_3\!: \ R_1[X(k)]\le 0 \cap R_2[X(k)]> 0 \cap R_3[X(k)]\le 0 \cap R_4[X(k)]\le 0 \cap \ldots\ R_M[X(k)]\le 0\\
&E_4\!: \ R_1[X(k)]> 0 \cap R_2[X(k)]> 0 \cap R_3[X(k)]\le 0 \cap R_4[X(k)]\le 0 \cap \ldots\ R_M[X(k)]\le 0\\
&E_5\!: \ R_1[X(k)]\le 0 \cap R_2[X(k)]\le 0 \cap R_3[X(k)]> 0 \cap R_4[X(k)]\le 0 \cap \ldots\ R_M[X(k)]\le 0\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.\\
&E_L\!: \ R_1[X(k)]> 0 \cap R_2[X(k)]> 0 \cap R_3[X(k)]> 0 \cap R_4[X(k)]> 0 \cap \ldots\ R_M[X(k)]> 0
\end{aligned}
\tag{4.1}
$$

representing specific combinations of classification rules satisfied by every vector $X(k)$. First, note that $L=2^M$. Probabilities of these events, evaluated separately for vectors X^a and X^b, constitute the following set of conditional probabilities instrumental for the decision making procedure:

$$P\{E_i / a\} \text{ and } P\{E_i / b\},\ i = 1,2,\ldots,L \tag{4.2}$$

These off-line tasks could be viewed as the development of a cluster model of the network and constitute the "learning" process.

Now consider the utilization of the cluster model for on-line attack detection. Note that at this point, we will consider any abnormal state of the network as a manifestation of the incipient attack. Assume that the probability of attack on the network has some initial value, established according to the existing statistics, $\gamma[0]$, and therefore the probability of normal operation, $\lambda[0] = 1 - \gamma[0]$.

Assume that vector $X(k) = \left[x_1(k), x_2(k), x_3(k), \ldots, x_n(k)\right]$ represents the most recent state of the network. Numerical values of components of this vector, applied to the classification rules $R_i[X],\ i=1,2,..,M$, results in a particular combination of numerical values

$$R_1[X(k)],\ R_2[X(k)],\ R_3[X(k)],\ \ldots,\ R_M[X(k)]$$

that could be identified as the occurrence of one of the events (4.1), for example, E_j. Now the availability of conditional probabilities (4.2) facilitates the application of Bayesian approach for the re-evaluation of the probability of the attack (i.e. the probability of the point *X(k)* to be marked by an "**a**") subject to the occurrence of the event E_j, $P\left\{a / E_j\right\}$. One can realize that unconditional probabilities, $\mathrm{P}\{\mathrm{a}\} + \mathrm{P}\{\mathrm{b}\} = 1$, therefore

$$P\left\{a / E_j\right\} P\left\{E_j\right\} = P\left\{E_j / a\right\} P\{a\} \text{ and } P\left\{E_j\right\} = P\left\{E_j / a\right\} P\{a\} + P\left\{E_j / b\right\} P\{b\},$$

and the required probability can be expressed as,

$$P\left\{a / E_j\right\} = \frac{P\left\{E_j / a\right\} P\{a\}}{P\left\{E_j / a\right\} P\{a\} + P\left\{E_j / b\right\} P\{b\}} = \frac{\gamma[0] \cdot P\left\{E_j / a\right\}}{\gamma[0] \cdot P\left\{E_j / a\right\} + \lambda[0] \cdot P\left\{E_j / b\right\}} \tag{4.3}$$

Computation (4.3) results in an updated value of the probability of attack, $P\left\{a / E_j\right\}$ that could be compared against some arbitrary defined threshold value. A warning message indicating an incipient attack should be issued if the probability of attack exceeds the threshold. This completes one cycle of the proposed procedure. At the next cycle, the "prior" probabilities of attack and the normal operation are defined as,

$$\gamma[k] = P\left\{a / E_j\right\} \text{ and } \lambda[k] = 1 - P\left\{a / E_j\right\}$$

and the new value of the network state vector,

$$X(k+1) = \left[x_1(k+1), x_2(k+1), x_3(k+1), \ldots, x_n(k+1)\right]$$

is to be analyzed with the consequent computation of probabilities $\gamma[k+1]$ and $\lambda[k+1]$. This procedure, intended for continuous real-time application, is capable of providing a timely and objective information to the network operator providing mathematically justified support of his/her decisions.

5. Definition of the "Normal" Status of a Computer Network

The definition of the "normal" status of a computer network depends on the particular configuration and patterns of usage of the specific network to be protected. For a given network, the availability of a network monitoring system allows us to accumulate a database, representing the network status observed during periods of both "normal" operation as well as specific attack scenarios. This database should provide sufficient information for the detection of specific changes in the network status caused by information attacks. Recall that each database record contains data representing the network status at a discrete time *k* and has the format {*X(k)*, *Q(k)*}, where *X(k)* is the state vector representing the status of the network at the discrete time *k* and *Q(k)* is a flag indicating the presence or absence of an attack at time *k*.

The dynamic properties of a computer network that represent its status are described by the properties of the network traffic flows through its particular modules. These properties include the volumes, compositions and rates of change of the traffic flows. Although these parameters fluctuate during normal periods of network operation, it is possible to distinguish a particular information attack based upon the unique effect it has on these parameters.

The experimental network chosen for this study is representative of many intranet configurations currently in use. The specific configuration of the experimental network is shown in Figure 8. It consists of a small local area network (LAN) with a connection to the Internet provided through a router, which lies outside of the administrative domain of the experimental setup. The connection to the router thus represents the "gateway connection" through which all external traffic enters the experimental network. This figure can be generalized to represent nearly any computer network. We have placed the network monitoring station at the gateway connection of the network to be able to easily monitor all traffic flows into the network.

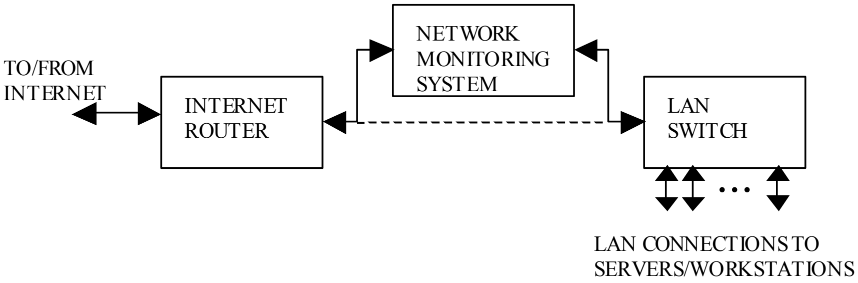

TO/FROM INTERNET
INTERNET ROUTER
NETWORK MONITORING SYSTEM
LAN SWITCH
LAN CONNECTIONS TO SERVERS/WORKSTATIONS

Figure 8. Experimental network configuration

Our experimental setup consists of a 100 Mbps switched Ethernet network interconnecting several SUN UltraSparc computers, some of which are client workstations while others serve as Web and file servers. The network monitoring systems consists of a SUN UltraSparc workstation with two network interface cards placed inline at the network gateway connection. This system monitors network packets both entering and leaving the network. Since the experimental setup uses TCP/IP protocols, the state space of the network has been defined in terms specific to these protocols. State variables represent measured quantities of the aggregate network traffic, using a variety of decompositions. For each measured quantity, its time derivative is also used as a state variable.

Information attacks usually attack a specific protocol. For example, TCP SYN attacks target the TCP protocol, while the Ping Flood attack targets IP control protocol. The aggregate network traffic is decomposed into streams identified by network protocol. The fractions of total packets utilizing the TCP and UDP protocols are measured.

The aggregate network traffic is then characterized by its general flow characteristics. The total volume of network traffic is measured, in bytes per second (normalized to the maximum rate allowed by the network). We also measure traffic rate in packets per second, (normalized to the maximum packet rate for the network). The average packet length may provide an indication of irregular packet flows that could be a sign that an attack is in progress. The traffic stream is decomposed into percentages of large, medium, and small packets. Without loss of generality, we have considered 100 bytes or less to be a small packet, and 500 bytes or greater to be a large packet; all others are considered medium packets.

Data Traffic Variables Subjected to Statistical Analysis

Variable	Description	Comment
x_1	fraction of packets utilizing TCP protocol	relative units
x_2	fraction of packets utilizing TCP protocol, rate of change	relative units
x_3	fraction of packets utilizing UDP protocol	relative units
x_4	fraction of packets utilizing UDP protocol, rate of change	relative units
x_5	fraction of large (>500 bytes) packets in the flow	relative units
x_6	fraction of large (>500 bytes) packets in the flow, rate of change	relative units
x_7	fraction of medium packets in the flow	relative units
x_8	fraction of medium packets in the flow, rate of change	relative units
x_9	fraction of small (<100 bytes) packets in the flow	relative units
x_{10}	fraction of small (<100 bytes) packets in the flow, rate of change	relative units
x_{11}	total volume of traffic, normalized	packets per sec
x_{12}	total volume of traffic, normalized, rate of change	relative units
x_{13}	fraction of TCP packets that are control type packets	relative units
x_{14}	fraction of TCP packets that are control packets, rate of change	relative units
x_{15}	fraction of IP packets that are control type packets	relative units
x_{16}	fraction of IP packets that are control type packets, rate of change	relative units
x_{17}	fraction of total packets that are control type packets	relative units
x_{18}	fraction of total packets that are control packets, rate of change	relative units
x_{19}	traffic volume, normalized	bytes per second
x_{20}	traffic volume, , rate of change	relative units
x_{21}	fraction of IP packets that are fragments	relative units
x_{22}	fraction of IP packets that are fragments	relative units
Q = ”A”	indication of a known attack	
Q = “B”	normal operation of the network	

When network messages are too large for network packets, the message is fragmented and transmitted into several packets. Some attacks, such as Ping Flood, may be characterized by a large number of fragmented messages. The ratio of fragmented traffic to total traffic is measured, as well as the rate of change of the ratio.

Most network protocols identify both data packets, which carry user application data, and control type packets, which are used to implement the protocol itself. Many attack scenarios exploit holes in the protocol architecture or bugs in a specific implementation of a protocol. Thus, the percentage of packets that are control type packets are measured for each protocol. In the TCP protocol, for example, we classify a packet as a control packet if the SYN, FIN, or RESET flags are set. We also measure the fraction of total traffic that represents control type packets. The above table summarizes designation of particular variables – components of the database.

6. Experimental results and their interpretation

To evaluate the proposed approach to attack detection in computer networks, we have chosen the two most common types of denial of service attacks, TCP SYN attack and Ping Flood attack, that were deployed in an experimental computer network. A software implementation of each attack was written in C++ and launched from computers outside of the experimental network. Attack packets were tagged (utilizing the unused type of service field in the IP packet header) for classification by the network monitoring station.

For each attack type, a database of network status vectors was collected over a 24 hour period at 1 second intervals. During this 24 hour period, two attacks were launched at arbitrary times. Thus, the database for each attack contained vectors representing both the normal status of the network and the status of the network during an attack.

TCP SYN Attack

The results of cluster analysis of the "attack/normal" data traffic are shown in Figs. 9.1-9.3. The informative components of the network status chosen by the cluster analysis procedure are fraction of small packets (x_9), fraction of control packets (x_{17}), fraction of TCP packets (x_1), and fraction of UDP packets (x_3). These were combined into the informative subspaces (x_9,x_{17}), (x_3,x_{17}), and (x_1,x_{17}). TCP SYN attacks are characterized by a large number of TCP control packets (SYN packets) flooding the target network. These packets are small (generally 40 bytes). Each figure shows the ellipse chosen by the cluster analysis procedure plotted against the data. In all cases, a point inside the ellipse is classified as an attack point.

From Figure 9.1, it is clear that during a TCP SYN attack there is both a high percentage of small packets and a high percentage of control packets, consistent with the nature of the attack. Since nearly all control packets are small, the fraction of control packets rarely exceeds the fraction of small packets, explaining the near absence of points above the line x_{17}=x_9. Attack points represent a high fraction of both control and small packets; as these fractions approach the maximum they are nearly equal and appear closely spaced around the line x_{17}=x_9. There are a few non-attack points that exhibit both a high fraction of control packets and a high-fraction of small packets. However, unlike a TCP SYN attack scenario, these points represent periods when the network was relatively quiet and thus generally will not be classified as attack points in the other subspaces. This illustrates the increased classification accuracy gained by considering several informative subspaces.

Figures 9.2 and 9.3 show the second and third most informative subspaces selected by the cluster analysis, respectively. Since TCP SYN attack is characterized by a high percentage of TCP control packets, it is counterintuitive that the subspace (x_3,x_{17}), representing the fraction of control packets vs. the fraction of UDP packets, is considered "more informative" than subspace (x_1,x_{17}), representing the fraction of control packets vs. the fraction of TCP packets. One possible explanation is given as follows. The subspace in Figure 9.3 indicates that during an attack there is both a high percentage of TCP packets and a high percentage of control packets in the network. This is a situation that can also occur normally, however, especially when the network is experiencing a very light load. Figure 9.2 illustrates that during an attack, the flood of TCP SYN packets causes a decrease in the flow of UDP packets due to the high rate of control packets consuming network bandwidth. This combination, however, may not occur as frequently during normal operation, even when the network is experiencing a light load. Thus, cluster analysis has the ability to infer subtle relationships between network status variables which may not be obvious and which may indeed be counterintuitive.

Figure 10 illustrates the application of the results of the cluster analysis to the on-line detection of network attacks. Here, the established classification rules are used to compute a sequence of events (4.1) based on the varying status of the network. These events are used to update the probability of attack. In the figure, time is shown in seconds and the reference time was arbitrarily chosen just prior to the start of the attack. The figure presents both the rate of

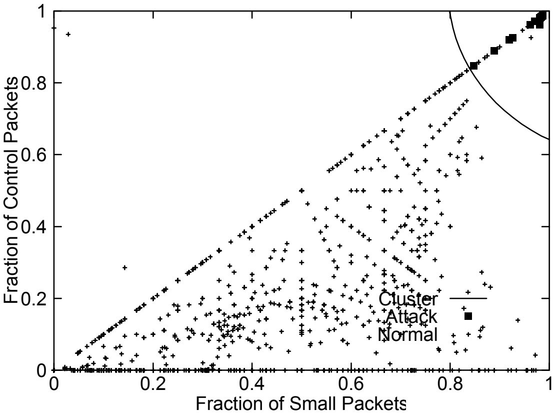


Figure 9.1

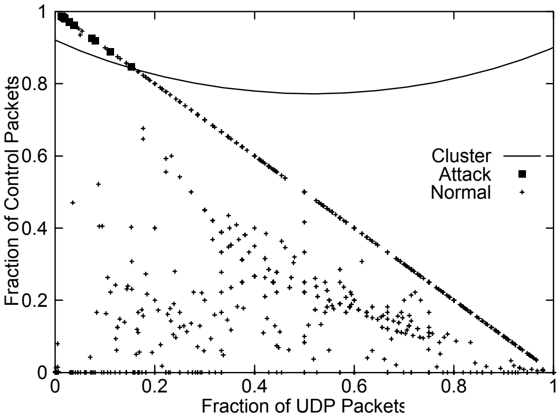


Figure 9.1

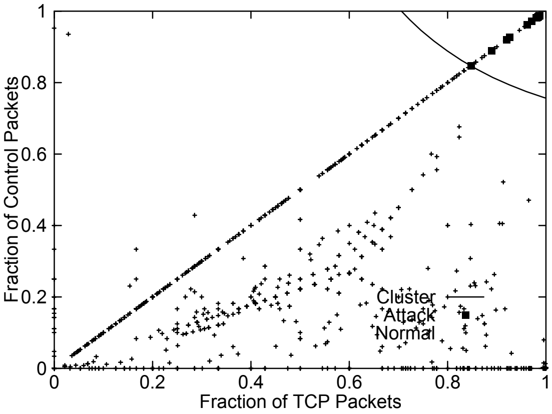


Figure 9.2

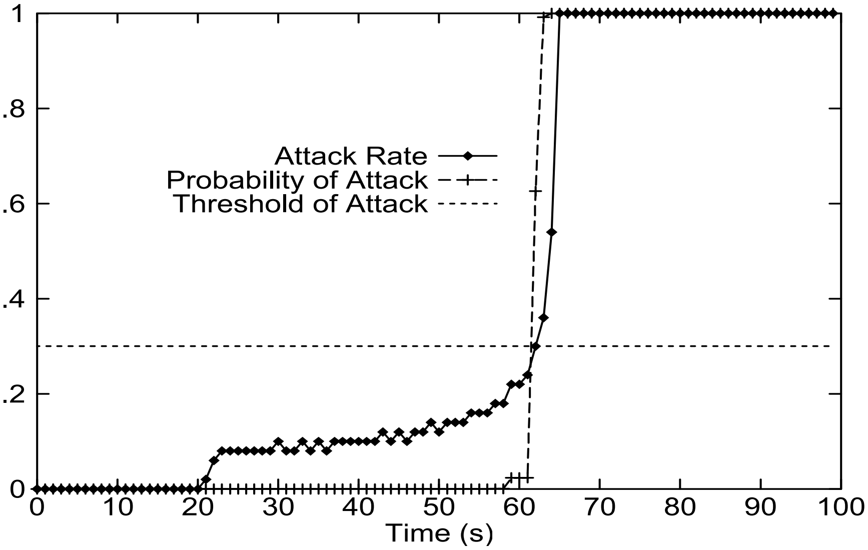


Figure 10. Attack Rate and Attack Probability

attack packets entering the network, normalized to the maximum rate observed, and the probability of attack iteratively calculated as per (4.3). This figure shows that the attack begins gradually, at about time 20sec, and increases in an exponential manner until it reaches its maximum rate. During the initial stages of the attack, the incoming traffic is light and does not have a significant effect on the probability of attack. At about time 58sec, the attack begins to increase in severity, and the probability of attack reacts quickly. Based on this data, we have set the "threshold of attack" at .3, at which point notification should be made in order that countermeasures be taken. It is interesting to note that the probability of attack reaches 1 before the attack reaches it full force, leading the attack and providing early warning. It should also be noted that the typical duration of a TCP SYN flood is measured in hours, while the proposed approach detected the attack in less that 40 s. The actual detection time is based on both the rate of progress of the attack and the normal traffic patterns on the network being protected.

Ping Flood Attack

The results of cluster analysis for Ping Flood on the experimental network are shown in Figs. 11.1-11.3. The informative components of the network status chosen by the cluster analysis procedure are fraction of IP control packets (x_{15}), fraction of control packets (x_{17}), rate of change in the fraction of IP control packets (x_{16}), and fraction of Large packets (x_5). These were combined into the informative subspaces (x_{17},x_{15}), (x_{16},x_{15}), and (x_{15},x_5). Ping Flood attacks are characterized by a large number of IP control packets (ICMP echo packets) flooding the target network. These packets are generally large.

From Figure11.1, it is clear that the most informative characteristics of a Ping flood attack are a large fraction of control packets, nearly all of which are IP control packets, consistent with the nature of the attack. Figure 11.2 shows that ping flood attacks can be distinguished from normal traffic that has a high fraction of IP control traffic based on the rate of change of IP control traffic. Figure 11.3 also illustrates that normal traffic can have a wide range of fraction of IP control traffic, but Ping Flood attacks can be distinguished by their high fraction of large packets in the network.

Figure 12 illustrates the application of the results of the cluster analysis to the on-line detection of ping flood attacks in the target network. This figure shows that the attack begins gradually, at about time 30s, and increases in an exponential manner until it reaches its maximum rate. Like the TCP SYN flood attack, during the initial stages of the attack, the incoming traffic is light and does not have a significant effect on the probability of attack. At about time 80s, the attack begins to increase sharply, and the probability of attack reacts. Again, the probability of attack leads the attack itself, reaching 1 before the attack reaches full force. Also, the attack was detected very early, in about 50 s in this experiment. Actual attacks are measured on a scale of hours.

It could be seen that an attack warning message could be generated should the attack probability value exceeds certain threshold. Otherwise, application of the developed software provides the network operator some important insights.

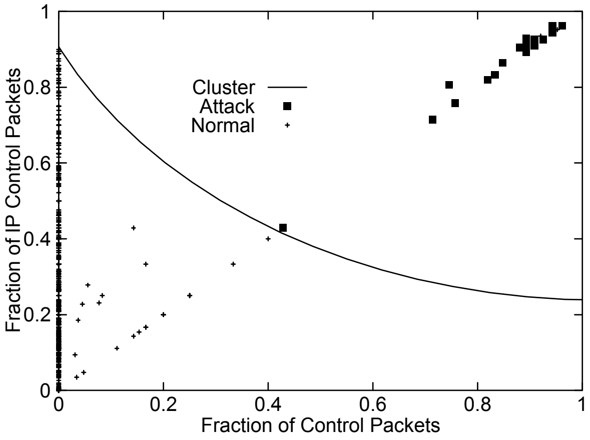


Figure 11.1

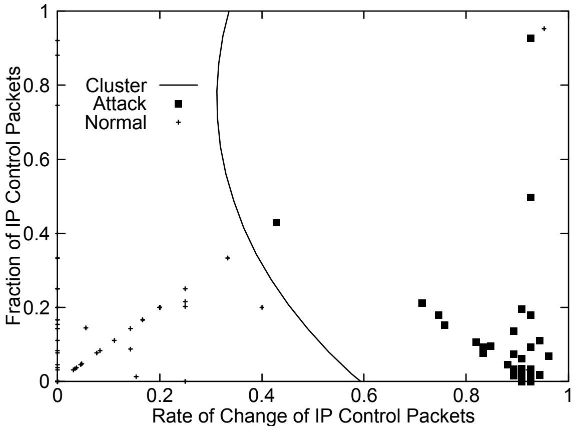


Figure 11.2

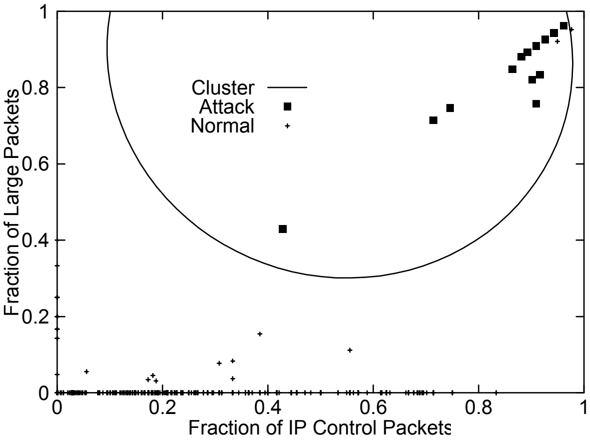


Figure 11.3

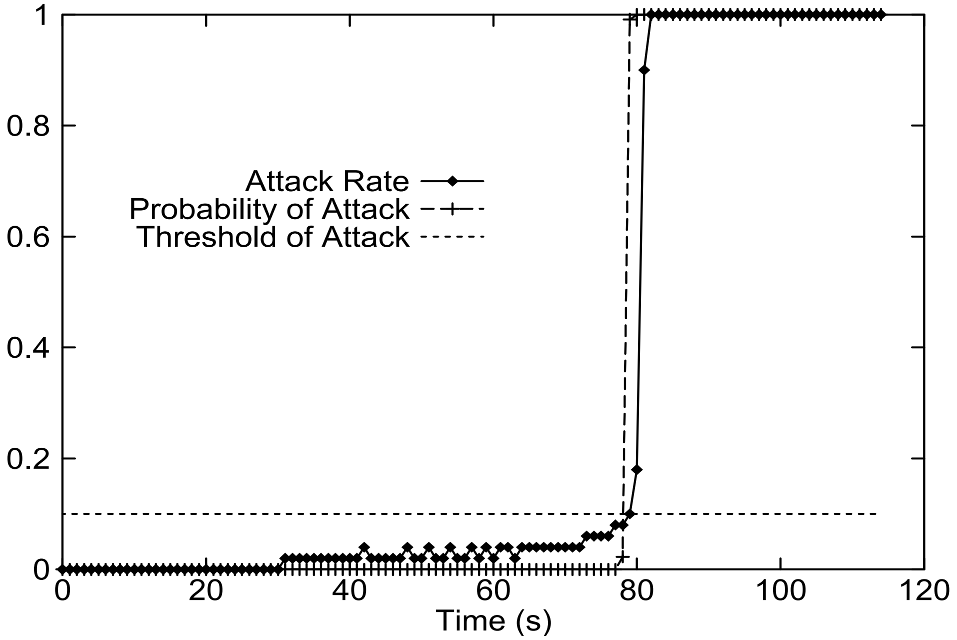


Figure 12. Attack Rate and Attack Probability

VIII. Experimental Computer Network Testbed

Overview

To conduct the experimental component of the project, a computer network testbed comprising several PCs and workstations, custom and commercial software, and imitating hosts and users is being developed. The configuration of this network has been chosen to be representative of many intranet configurations currently in use. The organization of the hardware configuration of the experimental network is shown in Figure X1. It consists of a small heterogeneous local area network (LAN) with a connection to the Internet provided through a router (not shown), which lies outside of the administrative domain of the experimental setup. To prevent backscatter and other malicious traffic from being unleashed onto the Internet, the experimental setup has only controlled and limited access to external networks. A gateway sentinel computer, which lies between the network access point and the external router, acts as a filter that removes malicious traffic *leaving* the network, while allowing malicious traffic to enter the network unrestricted. The network monitoring system measures the traffic before outgoing malicious traffic is removed, thus allowing the measurement of malicious traffic in both directions and providing the illusion of unrestricted access to the Internet. The monitoring and filtering functions have been designed to be implemented in very efficient software routines capable of providing the required database of network status records without impacting traffic flow.

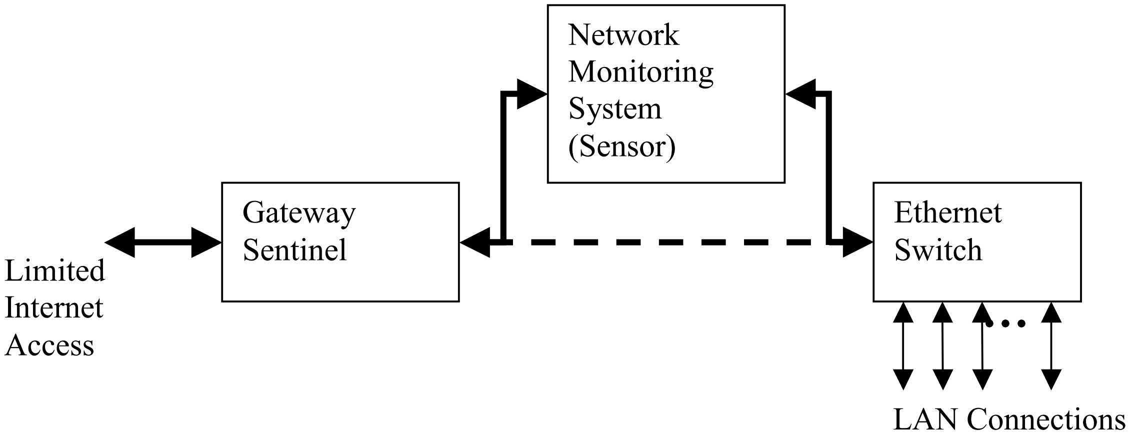


Figure 1. Experimental network configuration

The experimental setup consists of a heterogeneous switched Ethernet network interconnecting several Intel Pentium and SUN UltraSparc-based computers, some of which imitate client workstations while others act as Web and file servers. The network monitoring system consists of a Pentium-based workstation with two network interface cards, placed inline at the primary incoming network connection. This system monitors network packets both entering and leaving the network. Since the experimental network is representative of modern Internet Protocol based networks, standard off-the-shelf hardware and software components have been used.

To generate attack traffic, custom software has been written to emulate DoS attacks. For our initial investigation, the TCP SYN attack and the Ping Flood attacks have been chosen. These attacks were are the most common DoS attacks launched against internet servers. They are well-known, well understood, and easily implemented. By using these well-know attack scenarios, the behavior of the overall network can be more easily monitored. Other DoS attacks are being studied and will be tested.

Traffic Generation

Numerous databases of network status information have been collected from a live network to provide early data used for experimentation. Since the experimental network will be subjected to a number of malicious attacks, however, it is not reasonable to host any real network services, such as Web and file servers, on it. Thus, the normal flow of traffic on the experimental network will be stimulated using a combination of acquired and developed traffic models. These models will emulate the normal operation of the network and will contain synthetic traffic injected into the network as well as actual normal traffic generated locally. Synthetic traffic models will consist of actual measured packet traces gathered from a real network. The traffic generator will imitate external hosts and be responsible for the generation of all incoming traffic, while machines inside the test network will react and respond to this traffic (based on connection relationships and flows defined in the trace file). To maintain realistic traffic patterns, inter-packet timing relationships, obtained from the traces, will be maintained at the network hosts but packets in transit will be subjected to actual network delays. This means that host processing delays and "external" network delays, present in the traces, will be preserved, while network delays and inter-packet timing relationships will vary depending on the actual network operation. This should provide a reasonably realistic model of normal network traffic. As part of the proposed effort, this model will be evaluated against measured traffic patterns to determine it's accuracy in capturing network dynamics.

Once the normal traffic patterns are established, the network will be subjected to several information attacks. Using information obtained from CERT [20] and other sources on current trends and known models of DoS attacks, a number of well-known DoS attack scenarios will be implemented and launched. If available, traces from actual DoS attacks will also be used to generate DoS attack models.

Attack Generation

To generate attacks on the experimental network, custom attack programs have been written for the TCP SYN and Ping Flood attacks. We have initially chosen these two attacks because they are very easy for attackers to launch, are very effective at disabling the target, and attack the two most prevalent protocols used in the internet: Transmission Control Protocol and Internet Protocol.These programs simulate the occurrence of distributed DoS attacks. These programs are executed from the gateway sentinel and inject malicious traffic into the experimental network. The characteristics of the attack can be varied to simulate a number of different attack scenarios.

A TCP SYN flood attack is characterized by a large volume of transmission control protocol (TCP) connection initiation packets flooded into the target network. TCP is one of the main protocols used in the Internet. As part of the connection initiation procedure, TCP uses a three-way handshake. The client requests a connection with the server by sending a TCP packet with the synchronize (SYN) flag set. The server normally responds by replying with a packet that has both the SYN and acknowledgement (ACK) flags set, indicating that it is requesting a connection in the opposite direction and that it is acknowledging or accepting the original connection request. To complete the three-way handshake, the client responds with an ACK. In a TCP SYN flood attack, the attacker sends a large volume of TCP packets with the SYN flag set and a random or non-existent host in the return address field in the packet. When the packet is received by the server, the server reserves some memory for the connection and replies with the second part of the three-way handshake. Since the return address in the initial packet is not valid, however, the handshaking sequence never completes. Ultimately, the server runs out of memory to assign to new connections, preventing legitimate users from establishing new connections. In a distributed TCP SYN Flood attack, the attack is launched from many different computers, increasing the volume of packets flowing into the target network.

To simulate a distributed SYN flood, a freely available program called Neptune was obtained and modified to suit our needs. Neptune is a program that implements a TCP SYN flood attack from a single host. It periodically generates TCP SYN packets and sends them to the target with a return address specified as an input to the program. Neptune was not well suited for simulation of a distributed attack, which is generally implemented by controlling many slave computers from a master. The master sends a control to the slaves, indicating when the attack should begin. Since the slaves are not time synchronized and will not receive this command simultaneously, the attack will generally start off slowly and build in intensity as more and more slaves join in. To simulate a distributed attack, the timing characteristics of Neptune were modified. Packets are first sent at a predefined rate and the rate of attack is gradually increased as the attack progresses. We have implemented a number of methods for increasing the rate of attack, including linear, exponential, and random growth. Other major modifications were made to the program to suit our experimental environment, including the ability to tag attack packets and methods for preventing the reply packets generated at the target computer from leaving the test network. Our TCP SYN attack program allows us to alter the timing characteristics of the attack scenario, which allows training over a number of different scenarios as well as testing the attack detection software against previously unknown attack scenarios.

A Ping Flood attack exploits characteristics of the Internet Protocol (IP), the protocol that transfers packets across the Internet. A subset of IP, the Internet Control Message Protocol (ICMP), includes an echo request that can be used to determine if a computer’s IP software is functioning correctly. The client sends a packet requesting an ICMP echo and the server replies by returning the packet back to the client. By sending many such packets to a computer, and attacker can consume the bandwidth of the links connecting that computer to the network, preventing legitimate traffic from reaching the target. Another feature of IP that is often exploited in these attacks is fragmentation. When a very large packet is transmitted across the Internet, it often exceeds the maximum packet size supported by the intermediate network links over which it travels. In such cases, IP breaks the original packet into fragments, which are reassembled at the destination. By using the maximum IP packet size during a ping flood attack, the attacker can increase the volume of traffic to the target as well as consume resources at the target required to reassemble the packet.

A program has been written in C to simulate a distributed Ping Flood attack. This program opens a raw socket, which allows arbitrarily formed packets to be injected into the network. The program forms ICMP echo packets destined for the target network with pseudo random (or non-existent) source addresses. The program breaks large packets into fragments, which are subsequently put onto the network to be reassembled at the target. To simulate a distributed attack, the time between packet injections can be varied using a number of approaches, similar to those used in the TCP SYN attack software described above. Packets can be tagged for training purposes.

Network Monitoring System

The network monitoring system consists of a Linux-based workstation running a custom packet-sniffing program designed to capture relevant network status information. This program was written in C++ using the widely available PCAP packet capture library. With the network interface in promiscuous mode, all network packets are inspected and passed to a parsing function that analyses packet headers to compute relevant sensor statistics. Packets statistics are computed based on packet type and subtype, allowing collection of generalized network status as well as protocol specific information. The sensor periodically adds, at predefined intervals, a network status vector to a database. The sensor uses tags, placed in packets by the attack generation software, to classify each vector as attack or normal.

Since the experimental setup uses TCP/IP protocols, the state space of the network has been defined in terms specific to these protocols. Some state variables represent measured quantities of the aggregate network traffic, while others quantify individual components of network traffic, using a variety of decompositions. Changes in flow composition may provide an indication of irregular packet flows that could be a sign that an attack is in progress. Flooding-based DoS attacks, for example, are generally characterized by a large percentage of maximum-sized packets flowing into the network. These flow characteristics will be periodically written to the database, along with the rate of change of each. Some of the flow characteristics that will be monitored are described below.

The total instantaneous rate of the flow, in both bytes per second and packets per second, will be measured as well as the rates of change of the flow. The percentage of incoming and outgoing traffic will be calculated, and these will be further divided into percentages of small, medium, and large packets. When messages are too large for network packets, the message is fragmented and transmitted into several packets. Some attacks, such as Ping Flood, may be characterized by a large number of fragmented messages. The ratio of fragmented traffic to unfragmented traffic is measured, as well as the rate of change of this ratio.

Information attacks usually attack a specific network protocol. The aggregate network traffic is decomposed into streams, identified by network protocol. The percentages of packets tagged for a specific protocol, such as TCP, ICMP, IP, and UDP, are measured, as well as the derivatives of these rates.

For well-known protocol-specific attacks, a large occurrence of certain protocol commands or messages may indicate the presence of malicious activity. Protocol specific parameters to be measured include: the average rate of occurrence of TCP SYNs, TCP ACKs, and ICMP ECHOs (pings). The instantaneous rate of change of these quantities is also monitored.

Detection of a DoS attack at an earliest possible stage of its execution requires analysis of an appropriate quantitative representation of network status. In DoS attacks, single network packets do not operate in isolation. An attack consists of one or more streams of packets, aggregating into a flow of malicious traffic aimed at the target. Identification of individual streams that are part of an attack requires monitoring the state of packets in the network. Thus, attack detection requires both the observation of individual packets in the network as well as the combined effect of these packets on network state.

Table 1 shows the fields of individual packet headers that are monitored with the aim of differentiating legitimate packet streams from those that are part of an attack. The encoded bits of these fields can be treated as numerical values for purposes of numerical analysis. For a specific attack scenario, some parameters are likely to be more important than others. In some DoS attack scenarios, a specific network protocol is targeted. The protocol, port, and type fields contained in Internet packet headers provide information on the protocol service that the stream of packets is using. The sender of the packet is identified by the source address. Although the source address may be falsified to prevent identification of the attacker, some attack scenarios utilize a large number of unconcealed computers to flood the target, allowing separation of malicious traffic from innocuous traffic. The packet length and flow control information may provide an indication of irregular packet flows that could be a sign that an attack is in progress.

Table 1. Monitored Packet Parameters

Parameter Type	Parameters	Source	Justification
Target Protocol	Protocol Service Type	IP - protocol field TCP - port fields TCP - flags field	Some DoS attacks target a specific protocol.
Sender	Source Address	IP - source IP address	Attacks may be from unconcealed hosts.
Flow Information	Flow Control Packet Length	TCP - window size field IP - packet length field	Abnormal traffic patterns may be indicative of an attack.

Although individual flow parameters can be used to identify those packets that are part of an attack, it is necessary to monitor the status of the network in order to detect that an attack has begun. This includes parameters such as traffic volume and the number of certain types of connections that have been established in the network. In some attacks, the rate of change of these parameters may provide the earliest possible indication that an attack has begun. Some network status parameters that may be important to identifying a DoS attack are listed in Table 2.

An increase in the volume of traffic may indicate that a flooding type DoS attack has been launched. The volume of traffic (in bytes/s) and the packet rate in (packets/s) characterize the volume of traffic. Together, these two parameters also provide an indication of the average number of bytes per packet, which may be used as an indicator of abnormal traffic patterns. Flooding attacks aimed at a specific protocol are usually characterized by an inordinate amount of traffic using that protocol. The number of IP datagrams, ICMP messages, TCP segments, and UDP datagrams are monitored to detect such intrusions.

Some DoS attacks may establish a large number of connections the target network to exhaust the total number of connections. Others prevent legitimate access by leaving connections in an unusable state. Thus, the total number of connections is monitored, as well as those connections that are half-open (have only been partially established) or half-closed (are not in use, but have not been properly shut down). Finally, an attack might be indicated by an inordinate number of connection control operations. These include requests to reset connections as well as requests for retransmission of lost data.

Table 3 lists the components of the vectors that are produced by the network monitoring system. As part of this research, we are studying which characteristics of network traffic are most informative to permit detection of the widest range of DoS attacks with the smallest subset of components in the network status vector. Thus, components of the sensor vector are continually being updated.

Table 2. Aggregate Network State

Parameter Type	Parameter	Justification
Traffic Volume	Volume (bytes/s) Rate (packets/s)	Flooding type attacks may exhibit unusual volume or inordinate packet rate.
Protocol Usage	ICMP Packet Rate IP Packet Rate TCP Segments Rate UDP Datagram Rate	Protocol flood attacks may be characterized by high packet rates for the protocol.
Connection Status	Total Connections Total half-open Total half-closed	Some DoS attacks open a large number of connections, or leave many connections in a half-open state.
Connection Characteristics	Resets Sent Resets Received Retransmissions	Irregular traffic patterns may be indicated by large number retransmissions or resets.

Table 3. Components of network status vector produced by network monitoring system.

Variable	Description
x_1	volume of the flow, in bytes per second
x_2	volume of the flow rate of change
x_3	average length of a packet
x_4	average length of a packet rate of change
x_5	volume of the flow, TCP packets per second
x_6	volume of the flow, TCP packets, rate of change
x_7	volume of the flow, ICMP packets per second
x_8	volume of the flow, ICMP packets, rate of change
x_9	volume of the flow, IP packets per second
x_{10}	volume of the flow, IP packets, rate of change

x_{11}	volume of the flow, UDP packets per second
x_{12}	volume of the flow, UDP packets, rate of change
x_{13}	percentage of large packets in the flow
x_{14}	percentage of large packets in the flow, rate of change
x_{15}	percentage of medium packets in the flow
x_{16}	percentage of medium packets in the flow, rate of change
x_{17}	percentage of small packets in the flow
x_{18}	percentage of small packets in the flow, rate of change
x_{19}	average rate of occurrence of TCP SYNs
x_{20}	time derivative of the average rate of occurrence of TCP SYNs
x_{21}	average rate of occurrence of TCP ACKs
x_{22}	time derivative of the average rate of occurrence of TCP ACKs
x_{23}	average rate of occurrence of ICMP ECHOs
x_{24}	time derivative of the average rate of occurrence of ICMP ECHOs
Q = ”A”	indication of a known attack
Q = “B”	normal operation of the network

Real-Time Attack Detection Software

The real-time attack detection software is similar to the network monitoring software described above. This program is written in C++ using the PCAP packet capture library. With the network interface in promiscuous mode, all network packets are inspected and passed to the same parsing function, which computes relevant sensor statistics based on an analysis of packet headers. Instead of periodically adding the network status vectors to a database at predefined intervals, however, the vectors a reevaluated against the ellipse-based classification rules learned by the clustering algorithms. Based on the classification of each vector, an updated probability of attack is computed and output to the computer console.

To evaluate the proposed approach to attack detection in computer networks, we have chosen the two most common types of denial of service attacks deployed as of the writing of this paper: TCP SYN attack and Ping Flood attack. A software implementation of each attack was written in C++ and launched from computers outside of the experimental network. Attack packets were tagged (utilizing the unused type of service field in the IP packet header) for classification by the network monitoring station.

For each attack type, a database of network status vectors was collected over a 24 hour period at 1 second intervals. During this 24 hour period, two attacks were launched at arbitrary times. Thus, the database for each attack contained vectors representing both the normal status of the network and the status of the network during an attack.

The experimental results obtained using the network testbed are shown in the previous section of this Report.

IX. Further Research

The biological immune system is very effective in protecting a very complex organism from attacking organisms and dangerous proteins. It is automatic and requires no "user intervention". Extending functionality of the immune system to computer networks can potentially provide for a complete defense mechanism against malicious network traffic and computer code. The incidental similarities between both systems establish a solid foundation from which to build a more comprehensive network defense mechanism. The fundamental differences between the two is the lack of an "active" component in the computer system and coordination between existing defense mechanisms to assist in limiting or altogether preventing an attack.

It has been shown that an excessively long delay in the detection of the attack and limited system and network resources are factors that can lead to network fatality. It is apparent that the longer the delay of detection, the more resources are required to defend against the attack or that attempts to thwart an attack are unsuccessful. Since the DoS attack specifically attacks network resources, detection mechanisms that rely on previous known forms of attack or long-term detection are not suitable.

The following issues are to be investigated under the continuation of the BASIS project.

Establishment of accurate equations of attack/response dynamics in computer networks. As it was said earlier, equations representing the dynamics of the immune response may not be accurate due to poor knowledge of their parameters. Similar parameters representing a computer network and particular types of information attacks could be assessed with much greater accuracy. When established, these equations can be used for the definition of control laws to be implemented in computer security systems.

Distribution of Load. It has been shown that the effectiveness of thwarting an attack is determined by both the delay in the detection and the resource utilization of the system(s) under attack. Since modern network architectures are hierarchical, a single mechanism can be developed and deployed across a wide range of domains. Architecturally, a common model that captures the essential characteristics of networked computer systems could be established. Once modeled, a system of agents, which employ not only the networked computer systems, but also the networking hardware, can be developed to distribute the load for attack detection and suppression.

Self verses Non-Self. One of the hallmarks of the biological immune system is its ability to determine self from non-self. While, with computer software, there is a determining self/non-self aspect, with network traffic this determination is not so easy. Here the basic determination is malicious/non-malicious, or more generically normal/abnormal. There has been some research into this area, but none has yet been able to accomplish reliable and timely determination of malicious. We intend to continue this research, developing the established results further.

Failsafe system design. The designed defense mechanism should employ some functional redundancy. The biological immune system has many different levels of protection; each independently can either delay or prevent an attack from an antigen. Many computer defenses are singular, with only a single mechanism employed for each functional protection. This not only makes computer systems vulnerable to attack, the actual software employed can also be assaulted thereby crippling or disabling a system's defenses. We intend to develop practical, "biologically inspired" recommendations to computer network designers.

It was stated that one of the goals of the BASIS project is the cross-pollination between immunology and computer engineering. The following proposal recently funded by the AFOSR has been directly inspired by the BASIS research:

Recognition of Computer Viruses by Detecting their "Gene of Self-Replication" A high percentage of information attacks are performed by developing and distributing computer viruses over the Internet. Typically, the file containing a computer virus is transmitted over the Internet to the user's computer and subjected to decoding. Then, the interpreter generates the sequence of macro commands of the virus that are executed by the operating system thus resulting in the implementation of all destructive functions of the virus. Self-replication is a common feature of most computer viruses, and is quite uncommon for a legitimate code. As with any function of a computer code, self-replication is programmed, i.e. the sequence of macro commands resulting in the self-replication is present in the computer code of the virus. It is known that the function of self-replication could be implemented in many ways, and therefore, there is more than one sequence of macro commands that would perform this task. Moreover, it is expected that such a sequence is being dispersed within the entire body of the code and cannot be detected as an explicit pattern.

We are interested in the development of a methodology facilitating the detection of the "gene of self-replication" in computer codes. Unlike the existing anti-virus software, this methodology has the potential for providing the protection from previously unknown viruses. The feasibility of this task is justified by the following considerations:

- while the self-replication could be achieved in a number of different ways, this number is definitely finite
- by its nature, detection of one of the sequences of macro commands of interest in a large-size computer code is close to the problems of cryptology that can offer a number of successful techniques
- the problem has a straight-forward analogy, the detection of antigens by the immune system, and the detection mechanism could be adopted and replicated

We propose to conduct a feasibility study comprising the following tasks:

1. Establish a not necessarily complete set of alternative computer instructions implementing the task of self-replication. The resultant sequences of macro commands will be subjected to syntactic analyses leading to the most simplified and generalized description of the "gene of self-replication"
2. Investigate the self-replication techniques utilized in a number of known typical computer viruses
3. Develop, implement in software, and evaluate the efficiency of a number of techniques, adopted from cryptology and syntactic analysis, capable of detecting the required "word" in a randomly generated character sequence
4. Subject known typical computer viruses and legitimate software to syntactic/ cryptographic analyses aimed at the detection of the sequences responsible for the self-replication task. Assess performance of particular techniques and their potential for the development of on-line anti-virus software
5. Investigate the use of peptide-based antigen detection mechanism of the immune system and establish its formal description. Assess its potential for the development of on-line anti-virus software

X. References

Introduction

[1] S. Forrest, S. A. Hofmeyer, and A. Somayaji, “Computer Immunology,” *Communication of the ACM,* vol.40, No.10, pp.88-96, October 1997

[2] A. Somayaji, S. Hofmeyer, and S. Forrest, “Principles of a Computer Immune System,” *1997 New Security Paradigms Workshop,* pp. 75-82, Langdale, Cumbria, UK, 1997

[3] D. Dasgupta (ed), "Artificial Immune Systems and their Applications", Springer-Verlag, 1999

[4] M.Crosbie, E.Spafford. “Active Defending of a Computer System using Autonomous Agents,” *Technical Report No. 95-008. COAST Group,* Purdue University, 1995, pp.1-15

[5] J. Balasubramaniyan, J. Garcia-Fernandez, D. Isakoff, E. Spafford, and D. Zamboni, “An Architecture for Intrusion Detection using Autonomous Agents*,” In Proceedings of the 14th Annual Computer Security Applications Conference,* Phoenix, Arizona. December 7-11, 1998.

[6] V. I. Gorodetski, I. V. Kotenko, L. J. Popyack, and V. A. Skormin, “Agent-Based Model of Information Security System: Architecture and Framework for Behavior Coordination”, *Proceedings of the First International Workshop of Central and Eastern Europe on Multi-agent Systems (CEEMAS'99),* June 1999, pp. 323-331

[7] Skormin, V.A., Delgado-Frias, J.G., McGee, D.L., Giordano, J.V., Popyack, L.J., Gorodetski, V.I. and Tarakanov, A.O., BASIS: a biological approach to system information security, *Information Assurance in Computer Networks* (Gorodetsky V.I., Skormin V.A. and Popyack L.J. eds.., LNCS 2052, Springer-Verlag, Berlin, 2001, pp. 127-142).

[8] Tarakanov, V. Skormin, S. Sokolova, “Immunocomputing: Principles and Applications”, Springer-New York (in press)

Section III

[1] M.Crosbie, E.Spafford. “Active Defending of a Computer System using Autonomous Agents,” *Technical Report No. 95-008. COAST Group,* Purdue University, 1995, pp.1-15.

[2] Canadian Trusted Computer Product Evaluation Criteria, Canadian System Security Centre Communication Security Establishment, Government of Canada. Version 3.0e. January 1993.

[3] Common Criteria for Information Technology Security Evaluation. National Institute of Standards and Technology & National Security Agency (USA), Communication Security Establishment (Canada), UK IT Security and Certification Scheme (United Kingdom), Bundesamt fur Sichereit in der Informationstechnik (Germany), Service Central de la Securite des Systemes (France), National Communications Security Agency (Netherlands). Version 1.031.01.96.

[4] National Institute of Standards and Technology & National Security Agency, *Federal Criteria for Information Technology Security.* Version 1.0, December 1992.

[5] *Trusted Computer System Evaluation Criteria*. US Department of Defense 5200.28-STD, 1993.

[6] W. Brenner, R. Zarnekow, and H. Wittig, *Intelligent Software Agents: Foundations and Applications.* Springer-Verlag, 1998.

[7] S. Forrest, S. A. Hofmeyer, and A. Somayaji. "Computer Immunology," *Communication of the ACM,* vol.40, No.10, October 1997, pp. 88-96.

[8] S. Stainford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levit, C. Wee, R. Yip, and D. Zerkle. "GrIDS: A Graph-based Intrusion Detection System for Large Networks," *Proceedings of the 19th National Information System Security Conference,* Vol.1, National Institute of Standards and Technology, October, 1996, pp. 361-370.

[9] Hochberg et al., "NADIR: An Automated System for Detecting Network Intrusion and Misuse," *Computers and Security,* vol.12, No.3, 1993, pp. 235-248.

[10] G. White, E. Fish, and U. Pooch, "Cooperating Security Managers: A Peer-Based Intrusion Detection System," *IEEE Network,* January/February 1996, pp. 20-23.

Section IV

[1] Immunology Second Edition, Janis Kuby, W.H. Freeman and Company, New York, 1994

[2] Janeway, C.A., P. Travers, W. Walport, and J.D. Capra. 1999. Immunobiology. The immune system in health and disease. 4th Edition. Garland Publishing, New York

[3] Alberts, B., Bray, D., Lewis, I., Raff, M., Roberts, K. and Watson, I. D. 1994b. The immune system. In*: Molecular Biology of the Cell,* Third Edition, p. 1196, New York: Garland Publishing Inc.

[4] Avrameas, S. and Ternynck, T. 1993. The natural autoantibodies system: Between hypotheses and facts. *Molecular Immunology* 30, 1133-1142.

[5] Bona, C. A. 1988. V genes encoding autoantibodies: molecular and phenotypic characteristics. *Ann. Rev. Immunol.* 6, 327-358

[6] Burnet, F. M. 1957. A modification of Jerne's theory of antibody production using the concept of clonal selection. *Aust. J. Sci.* 20, 67-69

[7] D. Dasgupta (ed), "Artificial Immune Systems and their Applications", Springer-Verlag, 1999

Section V

[1] Immunology Second Edition, Janis Kuby, W.H. Freeman and Company, New York, 1994

[2] RFC 793 – Transmission Control Protocol, DARPA

[3] D. Dasgupta (ed), "Artificial Immune Systems and their Applications", Springer-Verlag, 1999

Section VI

[1] A. Tarakanov, V. Skormin, S. Sokolova "Immunocomputing: Principles and Applications", Springer (in press)

[2] D. Dasgupta (ed), "Artificial Immune Systems and their Applications", Springer-Verlag, 1999

[3] R.J. DeBoer, L.A. Segel and A.S. Perelson, "Pattern Formation in One and Two-Dimensional Shape Space Models of the Immune System", J. Theoret. Biol., Vol. 155, pp. 295-333, 1992

[4] N.K. Jerne, "Towards a Network Theory of the Immune System", Ann. Immunol. (Inst. Pasteur), Vol. 125C, pp. 373-389, 1974

[5] R. Horn and C. Johnson, "Matrix Analysis", Cambridge University Press, 1986

[6] V.I. Kuznetsov, A.F. Gubanov, V.V. Kuznetsov, A.O. Tarakanov and O.G. Tchertov, "Map of complex appraisal of environmental conditions in Kaliningrad", Kaliningrad: Ecological atlas (11 maps), 1999

[7] V.I. Kuznetsov, V.B. Milyaev and A.O. Tarakanov, "Mathematical Basis of Complex Ecological Evaluation", St.Petersburg University Press, 1999

[8] A.O. Tarakanov, "Mathematical Models of Biomolecular Information Processing: Formal Peptide Instead of Formal Neuron", Problems of Informatization J., Vol. 1, pp. 46-51, 1998 (in Russian)

[9] A. Tarakanov, "Formal Peptide as a Basic Agent of Immune Networks: From Natural Prototype to Mathematical Theory and Applications", Proc. of the 1st Int. Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS'99), pp. 281-292, St.Petersburg, Russia, 1999

[10] A. Tarakanov and D. Dasgupta, "A Formal Model of an Artificial Immune System", BioSystems, Vol. 55: 1-3, pp. 151-158, 2000

[11] A. Tarakanov, S. Sokolova, B. Abramov and A. Aikimbayev, "Immunocomputing of the Natural Plague Foci", Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2000), Workshop on Artificial Immune Systems, pp. 38-42, Las Vegas, USA, 2000

[12] A.O. Tarakanov, "Information Security with Formal Immune Networks", Information Assurance in Computer Networks (eds. V.I. Gorodetsky, V.A. Skormin and L.J. Popyack), pp. 115-126, Springer, 2001

[13] P. Wasserman, "Neural Computing. Theory and Practice", Van Nostrand Reihold, NY, 1990

Section VII

[1] V. Skormin and C. Herman, "Application of Statistical Clustering for Process Control of Screen Printing", ASME Electronic Packaging Journal, September 1995

[2] V. Skormin, V. Gorodetski, L. Popyack, "Application of Cluster Analysis in Diagnostics-Related Problems", Proceedings of the IEEE Aerospace Conference in Snowmass, CO, March 1999

[3] V. Skormin, L. Popyack and V. Gorodetski, "Data Mining Technology for Failure Prognostics of Avionics", IEEE Transactions on Aerospace and Electronic Systems, Vol. 38, No. 2, April 2002

[4] M. D. Vose, "The Simple Genetic Algorithm: Foundation and Theory," *MIT Press*, Cambridge, MA, 1999

[5] V. Skormin, V. Nikulin, and T. Busch, "Application of Genetic Algorithms for Optimal Design of Acousto-Optic Beam Steering Components," Optical Engineering, will appear in January 2002