# ADENOIDS: A HIBRID IDS BASED ON THE IMMUNE SYSTEM

*Fabrício S. Paula, Marcelo A. Reis, Diego A. M. Fernandes and Paulo L. Geus*

Computing Institute - State University of Campinas
Avenida Albert Einstein, 1251
Caixa Postal 6176, CEP 13083-970
Campinas-SP, Brazil
{*fabricio,marcelo.reis,diego.fernandes,paulo*}*@ic.unicamp.br*

**ABSTRACT**

The human immune system provides a rich source of inspiration for computer network security. By exploring this analogy, the authors propose a hybrid intrusion detection architecture that has the same learning and adaptive capability of the human immune system.

## 1. INTRODUCTION

From the viewpoint of traditional computer security, it is possible to guarantee the security of a computer system observing the following issues. It is necessary to correctly especify and implant a security policy, correctly design and implement the programs, and configure the system properly [1]. However, in practice, it is seen that security policies, programs' implementation, and systems' configuration might contain flaws that lead to an imperfect security [2, 8].

A higher security level can be achieved by adopting additional resources and design models that very closely resembles the conditions in which most computer networks currently exist—a hostile and prone to flaws environment. It is possible to find in nature a defense model that has many features that are desirable for a security system: the human immune system.

Once it is able to guarantee the survival of an individual for almost 70 years, even though he/she encounters potentially deadly parasites, bacteria and viruses in a daily basis, the immune system has a very strong analogy with computer network security.

This analogy between computer security problems and biological processes was first recognized in 1987, when Adelman [4] introduced the term "computer virus". The connection between immunology and computer security initiated in 1994 with publications [5, 7], resulting in a series of other works.

The initial researches were concentrated on isolated mechanisms of the immune system and how they could be applied to improve the security of a system [8]. More recent work started to consider the overall framework of the immune system as a design model for a security system, based on a set of organizing principles of the human defense system [8]. However, most of the researches concentrate on development of anomaly intrusion detection systems. This approach, however, explores only a portion of the framework provided by the immune system.

This paper proposes a new approach on development of intrusion detection systems (IDS) based on the human immune system framework. This new approach considers the fact that the human immune system has many features of misuse detection (pattern matching, memory of known attacks, more specific recognition, for example) in addition to those features of anomaly detection that have been explored in past researches (knowledge of what is normal and detection of what is different from normal).

In this way, the authors propose a hybrid IDS model, based on the framework of the immune system, that is capable of detecting and identifing an attack, elaborating a specialized response measure, and recovering from the attack. Besides that, the proposed model has the same learning and adaptive capability of the human immune system, and so it is able to react to unknown attacks and to improve its response under subsequent exposures to the same attack.

This paper presents an improvement on the authors' research presented in [3] and is organized as follows. Section 2 presents an overview of the human immune system. The proposed IDS is described in Section 3 and some analogies between the immune system and the proposed model are pointed in Section 4. Section 5 shows some aspects of the ADenoIdS IDS that is based on the proposed model. Finally, Section 7 composes some conclusions about the work presented in this paper.

## 2. IMMUNE SYSTEM OVERVIEW

It is impossible to understand how the immune system can be used as a design model for a computer defense system without an overview of its framework. This section presents the immune system basic structures and explains the immune response. The material for this overview is largely based on [8, 10, 11, 12].

The innate and adaptive systems compose the immune system. The innate immune system represents the first defense line and it is distinguished by its innate feature and limitted capacity to differenciate an infectious agent from another (non specific detection). It is also recognized by its primary and non specific response (most often insufficient). Among its main components there are the physical and chemical barriers, such as the skin, and cells known as phagocytes that survey the body for foreign substances.

On the other hand, the adaptative immune system is able to identify a particular pathogen, allowing a more efficient response. Besides that, it is able to "memorize" an infectious agent and to respond more vigorously to new exposures to the same pathogen. It is composed of lymphocytes (T cells and B cells) and antibodies.

At the heart of the system is the ability to recognize and respond to substances called *antigens*. In order to do this, the im-

Thanks to FAPESP agency for funding.

mune system must perform pattern recognition tasks to distinguish molecules and cells of the body (called *self*) from foreign ones (called *nonself*). This pattern recognition is performed by the reaction between antigens and proteins (called *receptors*) on the surface of immune system cells. Antigens are the patterns to be matched, and receptors are a type of complement of antigens. When an antigen binds to a receptor, a matching occurs and the immune response starts.

Phagocyte receptors can bind to a set of structurally related antigens and so, its detection is not specific. B cell receptors, which are produced in soluble form as antibodies, can bind directly to free antigen. On the other hand, T cell receptors do not bind intact and free antigen, rather, they react with cell surface *major histocompatibility complex* (MHC) molecules that display antigens fragments, called *peptides*. B and T cell receptors perform specific recognition of antigens.

The ability to detect most pathogens is partly achieved by generating receptors through a random process. However, only the receptors that do not bind to self proteins are choosen through a process called *negative selection*. During this process, recently created receptors are exposed to most self proteins; if any receptor binds to these self proteins it is eliminated. Negative selection creates the knowledge of what is normal (self) to the immune system[1].

The immune system has the ability to make its protection more specific by learning and memory[2]. If the immune system detects a pathogen that it has not encountered before, it undergoes a primary response, during which it "learns" the structure of the specific pathogen, evolving a set of its cells with high affinity for that pathogen, through a process called *affinity maturation*. On subsequent encounters with the same antigen pattern the immune system mounts a secondary response, using high affinity evolved cells retained in immune memory, that is more precise and efficient.

All immune cells and products (such as antibodies) circulate in the bloodstream, tissues and lymphatic vessels, acting as sentries on the lookout for foreign antigens. When receptors bind to antigens, on a sufficient concentration, an matching occurs and a complex set of events, called immune response, takes place resulting in the destruction of the infectious agents. The immune response can be splitted into three phases, as follows.

### 2.1. Detection Phase

When phagocytes or lymphocytes find foreign antigens they engulf and destroy them. After that, they display the antigen fragments combined with MHC molecules on their surface. If the foreign antigen is already "known" to the immune system, specific antibodies may bind directly to the antigen, making microbes attractive to other immune cells.

### 2.2. Antigen Presentation and Lymphocytes Activation Phase

Phagocytes and B cells that display MHC molecules with antigen peptides attracts circulating, resting T cells. If a T cell recognizes the antigen-protein complex and binds to it, it becomes activated and stimulates the transformation of the B cells into antibody-secreting plasma cells. Activated T cells start to reproduce and B cells start to produce specific antibodies—an antigen specific army is raised.

### 2.3. Antigen Elimination Phase

Specific antibodies bind to antigens marking them for destruction by phagocytes and cytotoxic T cells eliminate infected cells. As long as the concentration of foreign antigens decreases, all the chemical stimulus are gradually contained, leading to the immune response end. At the end, high affinity lymphocytes are retained in immune memory for future responses.

## 3. HYBRID IMMUNE BASED IDS MODEL

All researches on computer immunology, such as [6, 7], have focused on random generation of receptors and the process of negative selection of receptors that do not bind to self proteins. This approach is used in the quoted researches for the development of anomaly intrusion detection techniques. Basically they produce a database of what is considered to be normal in the system, and randomly generate receptors which are tested against the database of normal behavior. All receptors that fail to match any entry in that database is used to monitor the system, assuming that if it is activated, an abnormal situation has happened.

The new approach proposed in this paper is that the immune system also has some misuse intrusion detection features, and so, it represents a design model for a hybrid intrusion detection system. The immune memory is a database of signatures of known dangerous antigens, and antibodies and B cell receptors are signatures of specific antigens that the immune system has already encountered. These components of the immune system allow it to respond more efficiently to new exposures to a known invader. These are clearly misuse intrusion detection features, with an improvement—the immune system can autonomously change its misuse database (immune memory).

This section proposes an IDS model, based on the framework of the human immune system, that uses a hybrid architecture which applies both anomaly and misuse detection approaches [9]. Figure 1 illustrates this IDS model, presenting its components and the information flow between them. All its components are detailed as follows.

### 3.1. Data Source

The *data source* is responsible for collecting information and supplying a stream of event records to the *filtering system*. The nature of the information collected may vary according to the monitoring strategies adopted[3]: *host-based*, *network-based*, *aplication-based* or *target-based* [9]. The proposed IDS model is applicable to any of these strategies.

### 3.2. Filtering System

The *filtering system* provides audit reduction in order to identify and remove information that is redundant or irrelevant [9]. After filtering, the information stream is passed to the detection systems and, when required, to the *signature generator*.

[1] The knowledge of what is normal and detection of what is different from normal are anomaly intrusion detection features [9].

[2] The learning feature of the immune system relates to the learning of what is known to be "bad", and the immune memory is a sort of database of dangerous antigen signatures. These are features of misuse intrusion detection [9].

[3] It is assumed that the anomaly detection system may use a different monitoring strategy from the one adopted by the misuse detection system.

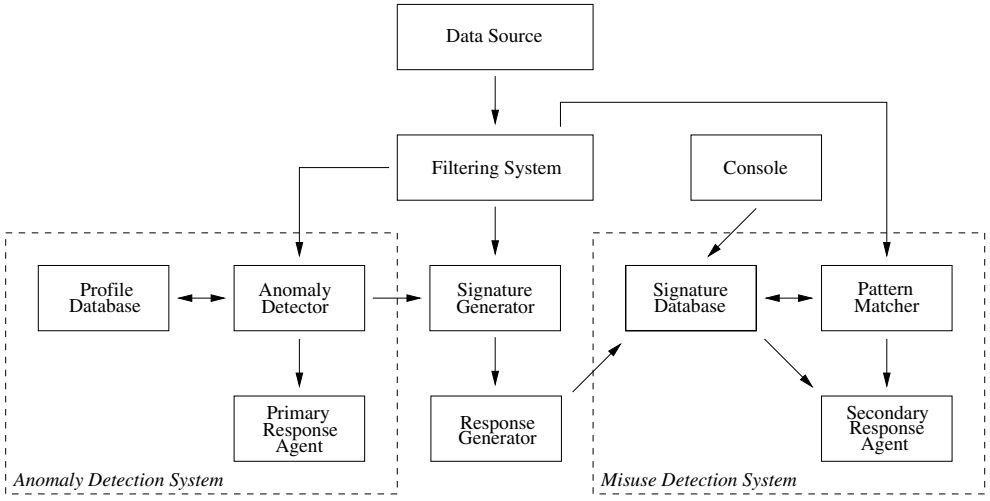


Figure 1: Hybrid immune based IDS model.

### 3.3. Anomaly Detection System

Anomaly detection involves a process of establishing profiles of normal behaviors, comparing actual behavior to those profiles, and flagging deviations from the normal (assuming it indicates misuse of the system). This approach accommodates adaptations to changes in normal behavior over time, adding learning and adaptability to the IDS [9]. The components of the *anomaly detection system* are described as follows.

#### *3.3.1. Profile Database*

The *profile database* is responsible for storing the profiles that describe the behavior of the computer system. These profiles may be traced through quantitative analysis techniques, statistical measures, neural networks, genetic algorithms and immune system approaches. Moreover, profiles are periodically and automatically updated to provide adaptive detection [9].

#### *3.3.2. Anomaly Detector*

The *anomaly detector* receives the event stream from the *filtering system* and verifies if it represents anomalous behavior. In order to do that, it compares the information received with the set of previously established profiles stored in the *profile database*. If any sign of abnormal behavior is detected, the *anomaly detector* activates the *primary response agent* and feeds the *signature generator* with the information detected as abnormal.

#### *3.3.3. Primary Response Agent*

Once activated, the *primary response agent* initiates a series of contention measures to slow down or even block a probable attack. The *primary response agent* reaction is limited and general once the attack is not specifically identified yet. The main purpose of these primary response measures is to minimize damage until a specific and efficient response can be executed. Some examples of such primary responses are: priority level reduction or process blocking, remote login disabling, filesystem protection and alarms of intrusive activities.

### 3.4. Signature Generator

An innovating feature of the proposed IDS is the convertion of information considered to be anomalous into a signature that specifically identifies the attack related to that abnormal behavior. This convertion introduces a learning capability, intrinsic to the anomaly detection, into the misuse detection system and provides a more efficient and precise detection of the attack in the future. In this way, the proposed IDS is able to automatically generate signatures of attacks that are unknown to the system. The *signature generator* is responsible for this convertion of anomalous information into a signature of the attack. After the generation of the signature, the *signature generator* activates the *response generator*.

### 3.5. Response Generator

The *response generator* receives the signature of the attack and elaborates a set of countermeasures specific to that attack. Both signature and response produced are delivered to the *signature database*.

### 3.6. Misuse Detection System

Misuse intrusion detection comprehends the search for activity patterns that match a known attack or other violation of security policy. This approach has shown to be efficient and reliable and, as a result, it is used on most commercial IDS [9]. The components of the *misuse detection system* are described as follows.

#### *3.6.1. Signature Database*

The *signature database* responsible for storing the signatures of attacks, relating them to the respective response measures. The signatures are used by the *pattern matcher*, while the countermeasures are consulted by the *secondary response agent*. In this way,

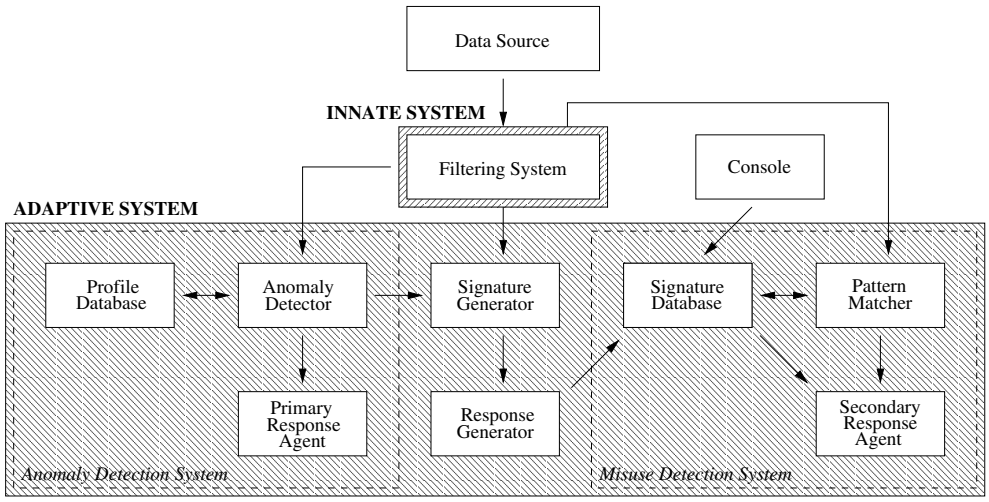


Figure 2: Analogy between innate, adaptive systems and the proposed IDS.

| IDS Components | Immune System |
|---|---|
| Data Source | Source of self and nonself proteins |
| Filtering System | Antigen presentation process |
| Profile Database | Set of random generated receptors |
| Anomaly Detector | Fagocyte non specific detection |
| Primary Response Agent | Innate system primary response |
| Signature Generator | Production of memory cells |
| Response Generator | Specific antibodies production |
| Signature Database | Set of high affinity memory cells |
| Pattern Matcher | Detection though memory cells |
| Secondary Response Agent | Specific immune response |
| Console | Artificially acquired immunity through vaccines |

Table 1: Analogies between the components of the proposed IDS and the immune system.

the proposed IDS can specifically detect and respond to each manifestation of a known attack in the system. The introduction of new signatures and countermeasures into the *signature database* can be conducted in two ways:

1. Automatically by the *response generator*;
2. Or manually by the system administrator through the *console*.

#### *3.6.2. Pattern Matcher*

The *pattern matcher* receives the event stream from *filtering system* and matches it with the patterns stored in the *signature database*. If any pattern is found in the event stream, the *pattern matcher* activates the *secondary response agent*. The detection is conducted in real time and uses an approach based on state transition [9].

#### *3.6.3. Secondary Response Agent*

Once activated, the *secondary response agent* receives the pattern that was matched and queries the *signature database* for the specific countermeasures related to that pattern. So the *secondary response agent* executes the countermeasures.

### 3.7. Console

The interface between the proposed IDS and the system administrator is possible through the *console*. This interface allows the inclusion and removal of signatures and countermeasures in the *signature database*.

## 4. ANALOGIES BETWEEN THE HUMAN IMMUNE SYSTEM AND THE IDS MODEL

As described in Section 2, the human immune system is divided into innate and adaptive systems. An analogy between these systems and the proposed IDS is illustrated on Figure 2.

The innate system is partially represented by the *filtering system* whose function resembles the process of antigen presentation. Other main features of the innate system, non specific detection and response, are present in the *anomaly detection system* (*anomaly detector* and *primary response agent* respectively),

which is properly modeled into the adaptive system considering the adaptive feature of the anomaly detection.

On the other hand, the adaptive system is represented by the components that implement learning and memory in the proposed IDS. Besides that, some of these components have other important features of the adaptive system, such as: accurate detection and efficient response.

Other analogies are presentated in Table 1, relating each component of the proposed IDS to the features of the human immune system.

## 5. ADENOIDS: AN IDS BASED ON THE PROPOSED MODEL

This section presents some details about the ADenoIdS[4] IDS. This IDS is an application of the model proposed in Section 3 to protect the computer system against buffer overflow attacks, that are considered the most important and persistent security problem [13, 14, 15]. It is assumed that the attackers do not have physical access to the hosts that they are attacking. In this way, the attacks must be remotely launched.

To face buffer overflow attacks, ADenoIdS adopt two monitoring strategies: host-based e network-based, ones achieved by an *anomaly detection system* and a *misuse detection system*, respectively. The *anomaly detector* performs a system call level detection, while the *misuse detector* analyses the network traffic related to the localhost. Therefore, each host to be protected must have an ADenoIdS IDS. The functioning of the IDS components are described as follows.

### 5.1. Data Source

The *data source* is responsible for collecting two types of information: system calls of any process and network traffic related to the localhost. Moreover, the *data source* must be able to store collected information temporarily in a log file, so that it can be consulted by the *filtering system*, if necessary.

### 5.2. Filtering System

The *filtering system* provides system call selection for the process specified by the *anomaly detector* or *signature generator*, and also network traffic according to filtering rules supplied by the *misuse detector* or *signature generator*. The *filtering system* can provide, for example, information such as "the last ten system calls by a process" or "the network traffic received by a process in the last minute".

### 5.3. Anomaly Detection System

An approach to achieve buffer overflow attack detection consists of analyzing, automatically or manually, the system calls by each process. This approach inhibits or even blocks an attack, before the system is compromised [16, 17]. The *anomaly detection system* is responsible for an automated detection of attack behavior. Its implementation is derived from [16], and its components are described as follows.

[4]In the human immune system, adenoids are specialized lymph nodes containing immune cells that protect the body against invaders of the respiratory system. We also use the term ADenoIdS for "**A**cquired **D**efense System Based on the **I**mmune **S**ystem".

#### 5.3.1. Profile Database

The *profile database* is responsible for storing, for each process, a system call profile that describes its normal behavior. These profiles are obtained through process execution analysis. A system call sequence that is not predicted in the process profile is interpreted as an attack sign.

#### 5.3.2. Anomaly Detector

The *anomaly detector*'s role is to verify if the system calls by each monitored process constitute an anomaly behavior. If any abnormal behavior sign is detected, the *anomaly detector* activates the *primary response agent* and feeds the *signature generator* with the system call sequence detected as abnormal.

#### 5.3.3. Primary Response Agent

Once activated, the *primary response agent* initiates a series of contention measures over the suspicious process, until the IDS or the system administrator can take an effective decision. Some contention measures can be: to insert process execution delays [16], block temporarily its execution and block the network traffic related to the process.

### 5.4. Signature Generator

The *signature generator* performs a sort of automated computer forensics. Based on the anomalous system calls received from *anomaly detector*, it initiates a search in the network stream log related to the suspicious process. In this way, the one intends to find signs of executable code that can generate the anomalous behavior previously detected. This detection of executable code can be helped by two techniques:

1. By searching for executable code patterns. For example, its possible to verify if there exists a code portion that performs a system call, as an assembly instruction 'int 0x80';
2. Making the search by packet replay. This technique consists in rebuilding the session according to the network traffic logged, in attempt to encounter the traffic portion that generate the detected anomaly.

This automated computer forensics approach is feasible because the attack has a remote origin. Therefore, if the attack is done, the anomalous code can only have arrived through the network traffic related to the attacked process.

Once identified a network traffic that produces the anomalous behavior, the *signature generator* creates an attack signature, that is delivered to the *response generator*. This signature is specified at the network level, so that it can be used by the *misuse detection system*. For example, a signature can express that an HTTP query holding a certain string (that is the suspicious executable code) represents an attack.

If the *signature generator* cannot find attack signs in the analyzed network traffic, the system administrator can be notified and the probable attack can be discarded, so much as the IDS is concerned. In this way, this analysis of atack signs also can contribute to minimize the false positive rate emitted by the *anomaly detector*.

### 5.5. Response Generator

The *response generator* receives the signature of the attack and elaborates a set of countermeasures, at network level, that deals in a specific way with the suspicious traffic. At the end, the signature and the countermeasures are inserted into the *signature database*, allowing the IDS to acquire immunity against the attack[5].

### 5.6. Misuse Detection System

The role of the *misuse detection system* is to perform detection and response in a real time fashion. When the *signature generator* produces a new attack signature, this signature is inserted into the *signature database*, with a response created by the *response generator*. It enables the IDS to detect and respond, at network level, against a new attempt of the same attack, before that code reaches up to the application level.

#### *5.6.1. Signature Database*

The *signature database* is responsible for storing the signature of attacks and their respective countermeasures. Each signature represents a network traffic that is considered an attack, and their countermeasures indicate what actions must be taken when this traffic is encountered.

#### *5.6.2. Pattern Matcher*

The specific detection is done by the *pattern matcher*, that intercepts the network traffic related with the localhost, searching for attack signatures stored in the *signature database*. If any attack pattern is found, the *pattern matcher* activates the *secondary response agent*.

#### *5.6.3. Secondary Response Agent*

Once activated, the *secondary response agent* performs the specific response related to the detected attack. This response occurs mainly at the network level.

### 5.7. Console

The *console* enables the system administrator to achieve insertion and deletion of signatures and responses from the *signature database*. In particular, if the IDS produces any improper signature or response, the *console* can be used to perform a correction.

## 6. ADENOIDS IMPLEMENTATION

The ADenoIdS implementation is based on Red Hat Linux, and consists of kernel patches and a set of tools. ADenoIdS is in a prototype phase and will be kept under the GNU General Public License.

## 7. CONCLUSION

The analogy between computer security and immunology represents a rich source of inspiration for development of new defense mechanisms, be it algorithms and intrusion detection techniques, security policies aware of possible flaws or even entire security systems. By exploiting this analogy, the proposed IDS combines learning and specialization into a hybrid architecture of intrusion detection and response. In this way, the proposed IDS is able of detect and respond to unknown attacks, improving its accuracy and efficiency on subsequent attacks.

[5] ... which justifies the "**A**cquired **D**efense" in the ADenoIdS name.

## 8. REFERENCES


[1] S. Garfinkel and G. Spafford. *Practical Unix and Internet Security*. O'Reilly, 1996. Second Edition.

[2] W. Venema. Murphy's Law and Computer Security. Presented in *Sixth USENIX Security Symposium*, San Jose, July 1996.

[3] M. A. Reis, D. M. Fernandes, F. S. Paula, and P. L. Geus. Modelagem de um Sistema de Segurança Imunológico. In *Proceedings of SSI'2001*, São José dos Campos, October 2001. (in Portuguese).

[4] F. Cohen. Computer Viruses. *Computers & Security*, 6:22-35, 1987.

[5] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonself Discrimination in a Computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamos, CA, 1994. IEEE Computer Society Press.

[6] S. A. Hofmeyr, and S. Forrest. Architecture for an Artificial Immune System. *Evolutionary Computation*, 7(1):45-68, MIT, 1999.

[7] J. O. Kephart. A Biologically Inspired Immune System for Computers. In *Artificial Life IV*, 1994. MIT Press.

[8] A. Somayaji, S. A. Hofmeyr, and S. Forrest. Principles of a Computer Immune System. In *Proceedings of the Second New Security Paradigms Workshop*, 1997.

[9] R. G. Bace. *Intrusion Detection*. Macmillan Technical Publishing, 2000.

[10] I. Roitt, J. Brostoff, and D. Male. *Immunology*. Mosby, 1996. Fourth edition.

[11] *Understanding Vaccines*. U.S. NIH Publication No. 98-4219, January 1998.

[12] *Understanding Autoimmune Diseases*. U.S. NIH Publication No. 98-4273, May 1998.

[13] C. Cowan, P. Wagle, C. Pu, S. Beattie and J. Walpole. Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade. In *DARPA Information Survivability Conference and Exposition*, January 2000.

[14] R. D. Pethia. Bugs in Programs. Keynote address at *SIGSOFT Foundations of Software Engeneering*, November 2000.

[15] B. Snow. Future of Security. Panel presentation at *IEEE Security and Privacy*, May 1999.

[16] A. Somayaji and S. Forrest. Automated Response Using System-Call Delays. In *Proceedings of the 9th USENIX Security Symposium*, August 2000.

[17] R. Sekar, T. Bowen and M. Segal. On Preventing Intrusions by Process Behavior Monitoring. In *Proceedings of the USENIX Workshop on Intrusion Detection and Network Monitoring*, April 1999.