Principles of a Computer Immune System

Anil Somayaji, Steven Hofmeyr, & Stephanie Forrest
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
{soma, steveah, forrest}@cs.unm.edu

Abstract

Natural immune systems provide a rich source of inspiration for computer security in the age of the Internet. Immune systems have many features that are desirable for the imperfect, uncontrolled, and open environments in which most computers currently exist. These include distributability, diversity, disposability, adaptability, autonomy, dynamic coverage, anomaly detection, multiple layers, identity via behavior, no trusted components, and imperfect detection. These principles suggest a wide variety of architectures for a computer immune system.

1 Introduction

Modern computer systems are plagued by security vulnerabilities. Whether it is the latest UNIX buffer overflow or bug in Microsoft Internet Explorer, our applications and operating systems are full of security flaws on many levels. From the viewpoint of traditional computer security, it should be possible to eliminate such problems through more extensive use of formal methods and better software engineering. We believe that such an approach is unlikely to succeed.

To see why, consider Figure 1a. This diagram is a slight caricature, but it does point out three key assumptions of the traditional view:

1. Security policy can be explicitly and correctly specified,
2. Programs can be correctly implemented, and
3. Systems can be correctly configured.

Although these statements might be true theoretically, in practice all are false. Consider Figure 1b. Computers are not static systems: vendors, system administrators, and users constantly change the state of a system. Programs are added and removed, and configurations are changed. Formal verification of a statically defined system is time-consuming and hard to do correctly; formal verification of a dynamic system is impractical. Without formal verifications, tools such as encryption, access controls, firewalls, and audit trails all become fallible, making perfect implementation of a security policy impossible—even if a correct policy could be devised in the first place.

Once we accept that our security policies, our implementations, and our configurations will have flaws, we must also accept that we will have imperfect security. This does not mean that we must be content with no security at all. As in the physical world, better security can be achieved with additional resources and better design. So, the real question is: how can we achieve better security than we currently have?

We believe it is possible to build better computer security systems by adopting design principles that are more appropriate for the imperfect, uncontrolled, and open environments in which most computers currently exist. As a case in point, we look to natural immune systems, which solve a similar problem, but in a radically different way from traditional computer security. For example, consider the human immune system. It is composed of many unreliable, short-lived, and imperfect components. It is autonomous. It is not "correct," because it sometimes makes mistakes. However, in spite of these mistakes, it functions well enough to help keep most us alive for 70+ years, even though we encounter potentially deadly parasites, bacteria, and viruses every day.

Some of the imperfections in current computer security are discussed in [15, 1]. The analogy between computer security problems and biological processes was recognized as early as 1987, when the term "computer virus" was introduced by Adelman [2]. The connection between immune systems and computer security was introduced in [7, 12] and elaborated in [6, 5]. However, in past work, we have concentrated on isolated ideas and mechanisms from the immune system and how they might be applied to concrete computer-security problems without explaining the overall framework. In this paper, we begin articulating the larger vision by discussing the immune system in terms of a set of organizing principles and possible architectures for implementation.

We believe that the success of the immune system is due

0

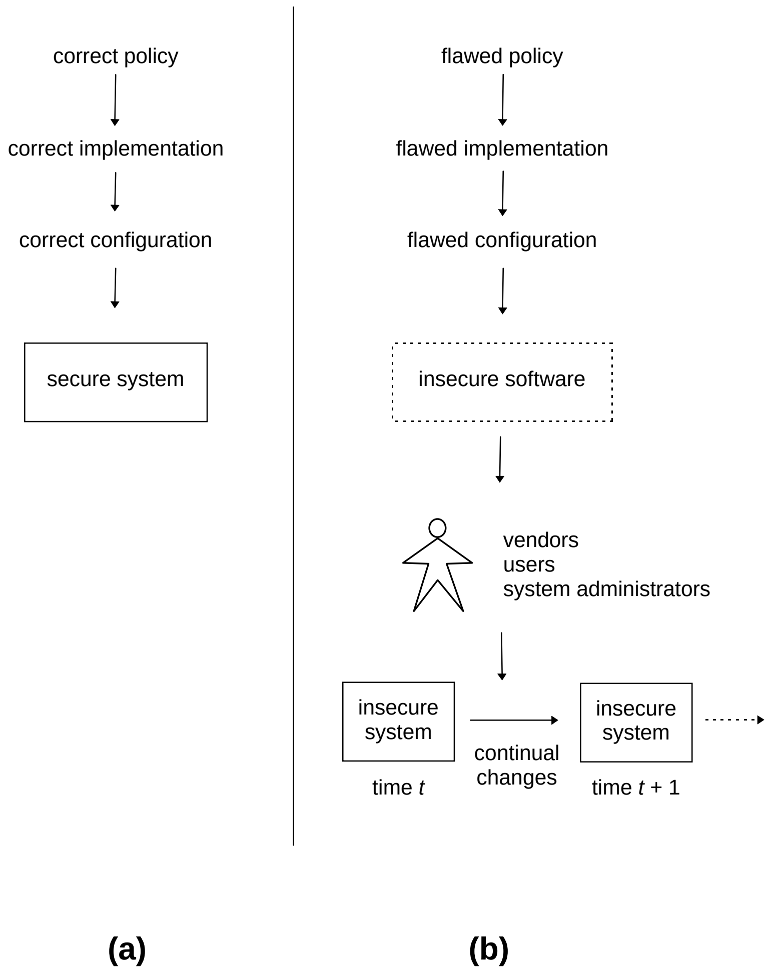

Figure 1: (a) Traditional view of secure systems development. (b) Real-world software development is an ongoing process, with vendor, system administrators, and users adding, modifying, and removing software continuously.

in large part to its organization and that an understanding of the immune system can help us design a robust, practical "computer immune system." Such a system would incorporate many elements of current security systems, augmenting them with an adaptive response layer.[1] Parts of this layer might be directly analogous to mechanisms present in the immune system; others will likely be quite different from those found in biology, even if they are based on similar principles to those found in the human body.

In the remaining sections of the paper, we first sketch how the human immune system works.[2] Then, we present a set of organizing principles that we argue accounts for much of the immune system's success. We also present some possible architectures for implementing computer security systems based on these principles. Finally, we discuss some limitations of the immune-system analogy.

2 Immune System Overview

The immune system defends the body against harmful diseases and infections. It is capable of recognizing virtually any foreign cell or molecule and eliminating it from the body. To do this, it must perform pattern recognition tasks to distinguish molecules and cells of the body (called "self") from foreign ones (called "nonself"). Thus, the problem that the immune system faces is that of distinguishing self from dangerous nonself. The number of foreign molecules that the immune system can recognize is unknown, but it has been estimated to be greater than 10^{16} [10]. These foreign proteins (kinds of molecules) must be distinguished from an estimated 10^5 different proteins of self, so recognition must be highly specific. These are staggering numbers, especially when one considers that the human genome, which encodes the "program" for constructing the immune system, only contains about 10^5 genes.

The architecture of the immune system is multilayered, with defenses provided at many levels. The outermost layer, the skin, is the first barrier to infection. A second barrier is physiological, where conditions such as pH and temperature provide inappropriate living conditions for some foreign organisms (pathogens). Once pathogens have entered the body, they are handled by the innate immune system and by the adaptive immune response. The innate immune system consists primarily of circulating scavenger cells such as macrophages that ingest extracellular molecules and materials, clearing the system of both debris and pathogens. The adaptive immune response (also called "the acquired immune response") is the most sophisticated and involves many different types of cells and molecules. It is called "adaptive" because it is responsible for immunity that is adaptively acquired during the lifetime of the organism. Because the adaptive immune system provides the most potential from a computer security viewpoint, we will focus on it in this overview. The material for this overview is largely based on [11]; we necessarily leave out many important details and emphasize the aspects most relevant to this paper.

The adaptive immune system can be viewed as a distributed detection system which consists primarily of white blood cells, called lymphocytes. Lymphocytes function as small independent detectors that circulate through the body in the blood and lymph systems. Lymphocytes can be viewed as *negative* detectors, because they detect nonself patterns, and ignore self patterns. Detection, or recognition, of nonself occurs when molecular bonds are formed between a pathogen and receptors that cover the surface of the lymphocyte. The more complementary the molecular shape and electrostatic surface charge between pathogen and lymphocyte receptor, the stronger the bond (or the higher the *affinity*). Detection is *approximate*; hence, a lymphocyte will bind with several different kinds of (structurally related) pathogens.

The ability to detect most pathogens requires a huge diversity of lymphocyte receptors. This diversity is partly achieved by generating lymphocyte receptors through a genetic process that introduces a huge amount of randomness. Generating receptors randomly could result in lymphocytes that detect self instead of nonself, which would then likely cause autoimmune problems in which the immune system attacks the body. Autoimmune disorders are rare because lymphocytes are self-*tolerant*, i.e. they do not recognize self. Tolerance of self is achieved through a process called clonal deletion: lymphocytes mature in an organ called the thymus through which most self proteins circulate; if they bind to these self proteins while maturing they are eliminated.

Even if receptors are randomly generated, there are not enough lymphocytes in the body to provide a complete coverage of the space of all pathogen patterns; one estimate is that there are 10^8 different lymphocyte receptors in the body at any given time [14], which must detect potentially 10^{16} different foreign patterns. The immune system has several mechanisms for addressing this problem, mechanisms which make the immune response more dynamic and more specific. Protection is made dynamic by the continual circulation of lymphocytes through the body, and by a continual turnover of the lymphocyte population. Lymphocytes are typically short-lived (a few days) and are continually replaced by new lymphocytes with new randomly generated receptors. Dynamic protection increases the coverage provided by the immune system over time: the longer a pathogen is present in the body, the more likely it is to be detected because it will encounter a greater diversity of lymphocytes.

Protection is made more specific by learning and memory. If the immune system detects a pathogen that it has not encountered before, it undergoes a primary response, dur-

[1]The adaptive response layer is similar in purpose to traditional intrusion-detection systems [4], although we are proposing a system that would be more autonomous.

[2]Although we describe the human immune system, other vertebrate immune systems are quite similar. Other natural immune systems, such as those of plants, have different architectures and mechanisms; however, they too have organizing principles similar to the human immune system.

ing which it “learns” the structure of the specific pathogen, i.e. it evolves a set of lymphocytes with high affinity for that pathogen, through a process called affinity maturation. This is a Darwinian process of variation and selection resembling the genetic algorithm. [9] High-affinity lymphocytes (those that bind most tightly with available pathogens) are stimulated to reproduce in great numbers, and the resulting lymphocytes have a large number of mutations. These new (mutated) lymphocytes then compete for pathogens with their parents and with other clones. Affinity maturation produces a large number of lymphocytes that have high affinity for a particular pathogen, which accelerates its detection and elimination. Speed of response is important in the immune system because most pathogens are replicating and will cause increasing damage as their numbers increase. Speed of response to previously encountered pathogens is generally high, because the information encoded in adapted lymphocytes is retained as immune memory. On subsequent encounters with the same antigen pattern the immune system mounts a secondary response. In this case, the adapted lymphocytes eliminate the pathogens so rapidly that the symptoms of the infection are not noticeable by the individual.

Even with all of these mechanisms, the coverage provided by the immune system is necessarily incomplete. The consequence is an immune system that is vulnerable to particular pathogens. However, not all individuals will be vulnerable to the same pathogens to the same degree, because each individual has a unique immune system. This diversity of immune systems across a population greatly enhances the survival of the population as a whole. One way in which immune systems differ from one individual to the next is by having different lymphocyte populations, and hence, different detector sets. Another key component that gives an immune system its uniqueness is the variation in a molecule called Major-Histocompatibility Complex (MHC). MHC molecules enable the immune system to detect intracellular pathogens (e.g., viruses) that reside inside cells. Intracellular pathogens are problematic because the inside of a cell is not “visible” to lymphocytes, that is, lymphocytes can only bind to structures on the surface of cells. MHC molecules bind to protein fragments called peptides (which could be viral) within a cell and transport the peptides to the surface, effectively displaying the contents of the cell to passing lymphocytes. The set of proteins to which an MHC molecule can bind is dependent on the structure of the MHC, which is genetically determined. Each person has only a limited number of MHC types and so is vulnerable to particular pathogens that cannot be readily transported by the available MHC types. However, as a whole, a population is far less vulnerable, because each individual has a different set of MHC types, and so is vulnerable to different pathogens.

To summarize, the natural immune system has many features that are desirable from a computer science standpoint. The system is massively parallel and its functioning is truly distributed. Individual components are disposable and unreliable, yet the system as a whole is robust. Previously encountered infections are detected and eliminated quickly, while novel intrusions are detected on a slower time scale, using a variety of adaptive mechanisms. The system is autonomous, controlling its own behavior both at the detector and effector levels. Each immune system detects infections in slightly different ways, so pathogens that are able to evade the defenses of one immune system cannot necessarily evade those of every other immune system.

3 Organizing Principles

Although the system described in the previous section is appealing, it is not immediately obvious how to use the immune system as a model for building successful computer security systems. There are several fundamental differences between the biology and computer systems. First, we desire an electronic system, built out of digital signals, not one constructed from cells and molecules. Further, we would like to avoid recreating all of the elaborate genetic controls, cell signalling, and other aspects of the immune system that are dictated by the physical constraints under which it evolved. Finally, the immune system is oriented towards problems of survival, which is only one of many considerations in computer security. Thus, the task of creating a useful system based on the immune-system analogy is a difficult one. In spite of these difficulties, a study of the immune system reveals a useful set of organizing principles that we believe should guide the design of computer security systems:

- *Distributability*: Lymphocytes in the immune system are able to determine locally the presence of an infection. No central coordination takes place, which means there is no single point of failure. A distributed, mobile agent architecture for security was also proposed in [3]. However, the human immune system provides a good example of a highly distributed architecture that greatly enhances robustness.

- *Multi-layered*: In the immune system, no one mechanism confers complete security. Rather, multiple layers of different mechanisms are combined to provide high overall security. This too is not a new concept in computer security, but we believe it is important and should be emphasized in system design.

- *Diversity*: By making systems diverse, security vulnerabilities in one system are less likely to be widespread. There are two ways in which systems can be diverse: the protection systems can be unique (as in natural immune systems and in [5]) or the protected systems can be diversified (as suggested in [8]).

- *Disposability*: No single component of the human immune system is essential—that is, any cell can be replaced. The immune system can manage this because cell death is balanced by cell production. Although we do not currently have self-reproducing hardware, death and reproduction at the process/agent level is certainly possible and would have some advantages if it could be controlled.

- *Autonomy*: The immune system does not require outside management or maintenance; it autonomously classifies and eliminates pathogens, and it repairs itself by replacing damaged cells. Although we do not expect (or necessarily want) such a degree of independence from our computers, as network and CPU speeds increase, and as the use of mobile code spreads, it will be increasingly important for computers to manage most security problems automatically.

- *Adaptability*: The immune system learns to detect new pathogens, and retains the ability to recognize previously seen pathogens through immune memory. A computer immune system should be similarly adaptable, both learning to recognize new intrusions and remembering the signatures of previous attacks.

- *No secure layer*: Any cell in the human body can be attacked by a pathogen—including those of the immune system itself. However, because lymphocytes are also cells, lymphocytes can protect the body against other compromised lymphocytes. In this way, mutual protection can stand in for a secure code base.

- *Dynamically changing coverage*: The immune system makes a space/time tradeoff in its detector set: it cannot maintain a set of detectors (lymphocytes) large enough to cover the space of all pathogens, so instead at any time it maintains a random sample of its detector repertoire, which circulates throughout the body. This repertoire is constantly changing through cell death and reproduction.

- *Identity via behavior*: In cryptography, identity is proven through the use of a secret. The human immune system, in contrast, does not depend on secrets; instead, identity is verified through the presentation of peptides, or protein fragments. Because proteins can be thought of as “the running code” of the body, peptides serve as indicators of behavior. We have proposed a computer analog to this, short sequences of system calls [6].

- *Anomaly detection*: The immune system has the ability to detect pathogens that it has never encountered before, i.e. it performs anomaly detection. We believe that the ability to detect intrusions or violations that are not already known is an important feature of any security system.

- *Imperfect detection*: By accepting imperfect detection, the immune system increases the flexibility with which it can allocate resources. For example, less specific lymphocytes can detect a wider variety of pathogens but will be less efficient at detecting any specific pathogen.

- *The numbers game*: The human immune system replicates detectors to deal with replicating pathogens. It must do so—otherwise, the pathogens would quickly overwhelm any defense. Computers are subject to a similar numbers game, by hackers freely trading exploit scripts on the Internet, by denial-of-service attacks, and by computer viruses. For example, the success of one hacker can quickly lead to the compromise of thousands of hosts. Clearly, the pathogens in the computer security world are playing the numbers game—traditional systems, however, are not.

These properties can be thought of as design principles for a computer immune system. Many of them are not new, and some have been integral features of computer security systems; however, no existing computer security system incorporates more than a few of these ideas. Although the exact biological implementation may or may not prove useful, we believe that these properties of natural immune systems can help us design more secure computer systems.

4 Possible Architectures

One approach to building computer security architectures that incorporate the principles discussed in the previous section is to design systems based on direct mappings between immune system components and current computer system architectures. A few such possibilities are described below.

- *Protecting Static Data*: A natural place to begin is at the level of computer viruses, which typically infect programs or boot sectors by inserting instructions into program files stored on disk. Under this view, the protection problem is essentially the same as that of protecting any kind of stored data—self is interpreted as uncorrupted data and nonself is interpreted as any change to self. Many change-detection algorithms have been devised to address this problem, including some inspired by biology [7]. Kephart has developed an architecture for protecting against viruses in a networked environment [12].

- *Protecting Active Processes on a Single Host*: The adaptive human immune system is made primarily out of cells which monitor and interact with other cells. If we view every active process in a computer as a cell, we can then think of a computer running multiple processes as a multicellular organism, and a set of networked computers as a population of such organisms. Traditional security mechanisms, such as passwords, groups, file permissions, etc., would play a role analogous to that of a computer’s skin and innate immune system. To create an adaptive immune system layer, we could implement a “lymphocyte” process which, with help from the kernel, is able to query other processes, to see whether they are functioning normally. Just as in the natural immune system, we assume that if a process is acting abnormally, it is either damaged or under attack. In response, the lymphocyte process could slow, suspend, kill, or restart the misbehaving process. To complete the picture, each lymphocyte process could have a randomly-generated detector or set of detectors, living for a limited amount of time, after which it would be replaced by another lymphocyte. This is important because it means that there would be no predefined location or control thread at which the protection system could be attacked. Lymphocytes that proved particularly useful during their lifetime (e.g, by detecting new anomalies) could be given a longer life-span or allowed to spawn related processes. Additionally, autoimmune responses (e.g., false alarms) could be prevented through a censoring process (analogous to clonal deletion in the thymus).

In this architecture, self would be defined by normal behavior and nonself would be abnormal behavior in the form of intrusions, either in privileged or in user processes. Such a system could adapt to changes in user behavior and system software through the turnover of lymphocytes (also making it vulnerable to “training” by malicious users). The level of security could be tuned by adjusting the number and lifetime of the lymphocytes, and by adjusting the number and quality of detectors in the lymphocytes.

In order to implement this architecture, however, we need an analog for peptide/MHC binding, and a mechanism for eliminating self-reactive detectors. We have already worked on the former: in [6] we examine approximate matching of short sequences of system calls as a candidate for distinguishing normal and abnormal behavior. A method for tolerance, and a complete implementation, are subjects of future work.

- *Protecting a Network of Mutually Trusting Computers*: Another approach is to think of each computer as corresponding to an organ in an animal. Each process would still be considered as a cell, but now an individual is a network of mutually trusting computers. In this model, the innate immune system is composed of host-based security mechanisms, combined with network security mechanisms such as Kerberos [13] and firewalls. The adaptive immune system layer could be implemented by kernel-assisted lymphocyte processes, with the added feature that these lymphocytes could migrate between computers, making them mobile agents. One computer (or a set of computers) could then be reserved as a thymus for the network, selecting and propagating lymphocytes, each of which searches for a specific pattern of abnormal behavior. If these lymphocyte processes use negative detection, no centralized server is needed to coordinate a response to a security breach; the detecting lymphocyte can take whatever action is necessary, possibly replicating and circulating itself to find similar problems on other hosts.[3]

  This architecture is similar to the previous one, except for the addition of circulating mobile detector processes. In principle, it should be able to detect the same class of anomalies. However, anomalies found on one computer could also be quickly eliminated from other computers in the network. It has similar requirements as before, except that it also depends upon a robust mobile agent framework. Because lymphocytes are also processes, they will monitor each other, ameliorating the dangers of rogue self-replicating mobile lymphocytes.

- *Protecting a Network of Mutually Trusting Disposable Computers*: Moving the analogy up another level, we could regard each computer as a cell, with a network of mutually-trusting computers being the individual. Host-based security can be thought of as the normal defenses a cell has against attack. The innate immune system consists of the network’s defenses, such as Kerberos and firewalls. We can implement an adaptive immune system layer by creating a set of lymphocyte machines. These machines would monitor the state of other machines on the network. When an anomaly was detected, the problematic machine could be isolated (perhaps by reconfiguring hubs and/or routers), rebooted, or shut down. If the true source of the anomaly were outside the network, a lymphocyte could stand in for the victimized machine, doing battle with the malicious host, potentially sacrificing itself for the good of the network.

  This architecture could address problems of compromised hosts, network flooding denial-of-service attacks, and even hardware failures. However, it has significantly more requirements than the previous two. An implementation would need an MHC/peptide analog at the host level, potentially based on a machine’s network traffic, or based on the behavior of its kernel. A dynamically configurable network topology would be necessary to allow lymphocyte machines to isolate a given host. As before, a thymus-type mechanism would be needed to prevent autoimmune responses. In particular, though, an implementation would require that most hosts be somewhat interchangeable—otherwise the network could not afford the loss of any hosts.

5 Limitations

Although we believe it is fruitful to translate the structure of the human immune system into our computers, ultimately

[3]This mechanism can be seen as a generalization of the kill-signal described in [12].

we are not interested in imitating biology. Not only might biological solutions not be directly applicable to our computer systems, we also risk ignoring non-biological solutions that are more appropriate. A more subtle risk, however, is that through imitation we might inherit inappropriate "assumptions" of the immune system.

Computer security is supposed to address five issues: confidentiality, integrity, availability, accountability, and correctness. In the immune system, however, there is really only one important issue, survival, which can be thought of primarily as a combination of integrity and availability. If we view immune system memory as a type of audit trail, it might be possible to argue that there is also a form of accountability, but it clearly is not the same kind of accountability that we typically associate with computer security. Correctness and confidentiality are largely irrelevant to survival. By correctness, we generally mean that it can be proved that a certain program meets its specifications. Immune systems are not formally specified systems, so by definition they cannot be called correct (in the formal sense). If we think of the environment in which an organism evolves as an implicit formal specification of "survival," it is still true that natural immune systems are not correct, because they sometimes fail—pathogens sometimes successfully evade the immune system. Likewise, the immune system is not concerned with protecting secrets, privacy, or other issues of confidentiality. This is probably the most important limitation of the analogy, and one that we should keep in mind when thinking about how to apply our knowledge of immunology to problems in computer security.

6 Conclusions

Good passwords, appropriate access controls, and careful design are still needed for good security. As indicated earlier, all of these measures can be seen as equivalent to the body's skin and innate immune system, which are responsible for preventing most infections. We have focused on the human immune system's adaptive responses, because these are the types of mechanisms current computer systems do not have. By remedying this shortcoming, we should be able to make our computer systems much more secure than they currently are.

7 Acknowledgments

The authors wish to thank David Ackley for his insightful suggestions and inspiring aphorisms that have helped us focus our ideas. Also, Alan Perelson has introduced us to the world of immunology, and Gene Spafford has helped us understand the finer points of computer security.

The authors gratefully acknowledge support from the National Science Foundation (grant IRI-9157644), the Office of Naval Research (grant N00014-95-1-0364), Defense Advanced Research Projects Agency (grants N00014-96-1-0680 and N66001-96-C-8509), the MIT Artificial Intelligence Laboratory, Interval Research Corp., and the Santa Fe Institute.

References

[1] R. Blakley. The emperor's old armor. In *Proc. New Security Paradigms '96*. ACM Press, 1997.

[2] Fred Cohen. Computer viruses. *Computers & Security*, 6:22–35, 1987.

[3] Mark Crosbie and Gene Spafford. Active defense of a computer system using autonomous agents. Technical Report 95–008, Department of Computer Science, Purdue University, February 1995.

[4] D. E. Denning. An intrusion detection model. In *IEEE Transactions on Software Engineering*, Los Alamos, CA, 1987. IEEE Computer Society Press.

[5] S. Forrest, S. Hofmeyr, and A. Somayaji. Computer immunology. *Communications of the ACM*, (submitted Dec. 1996).

[6] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff. A sense of self for UNIX processes. In *Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy*. IEEE Press, 1996.

[7] 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.

[8] S. Forrest, A. Somayaji, and D. H. Ackley. Building diverse computer systems. In *Sixth Workshop on Hot Topics in Operating Systems*, 1997.

[9] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, 1992. Second edition (First edition, 1975).

[10] J. K. Inman. The antibody combining region: Speculations on the hypothesis of general multispecificity. In G. I. Bell, A. S. Perelson, and Jr. G. H. Pimbley, editors, *Theoretical Immunology*, pages 243–278. M. Dekker, NY, 1978.

[11] C. A. Janeway and P. Travers. *Immunobiology: the immune system in health and disease*. Current Biology Ltd., London, 2nd edition, 1996.

[12] J. O. Kephart. A biologically inspired immune system for computers. In R. A. Brooks and P. Maes, editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation*

of Living Systems, pages 130–139, Cambridge, MA, 1994. MIT Press.

[13] B. C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38, September 1994.

[14] S. Tonegawa. Somatic generation of antibody diversity. *Nature*, 302:575–581, 1983.

[15] W. A. Wulf, C. Wang, and D. Kienzle. A new model of security for distributed systems. Technical Report CS-95-34, University of Virginia, August 1995.