Engineering an Immune System[1]

Stephanie Forrest
Dept. of Computer Science,
University of New Mexico
Albuquerque, NM 87131

Steven Hofmeyr
Dept. of Computer Science,
University of New Mexico
Albuquerque, NM 87131

Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501

February 11, 2001

[1]Submitted to Graft February 11, 2001

1. Introduction

The immune system is highly complex, and many researchers believe its complexity prevents us from ever knowing definitively what purpose each individual component serves. As they point out, the forces of natural selection don't guarantee perfect or minimal solutions, simply ones that work well enough to increase survivability. We know that many simpler immune systems exist in lower animals, and these systems seem to work just fine. Thus, so the argument goes, the human immune system is not optimized, and ascribing specific purpose to the various components and mechanisms is foolhardy.

Even worse, there is still debate about what role the immune system plays for the body. The traditional view that the immune system's primary job is to distinguish *self* from *nonself*[1] has been challenged in many places (e.g., [9]). Likewise, the view of the immune system as a "danger detector" is controversial (e.g.,[11]). And, more recently, some have advocated viewing the immune system as a means of preserving homeostasis for the body (e.g., [2]). These three views are not mutually exclusive, but they emphasize different aspects of immune-system function and suggest not only that the system itself is complex, but so is its functional role in the body. By contrast, there is little debate that the primary function of the heart is to circulate blood. In spite of these controversies, it seems reasonable to assume that one of the immune system's main roles in the body is "protective."

If we don't know exactly what the immune system does, and if we believe that many of its components are redundant or unnecessary, then how should we go about understanding the immune system? The point of this article is to suggest that an engineering perspective might be helpful. That is, if we set out to engineer a protective system that operates successfully in an environment with some of the same constraints as those faced by the immune system, what components would we need and what would we need them for? To what extent would these components resemble the natural immune system?

2. Building a computer immune system

A suitable domain for such an exercise is that of computer security. A computer security system should protect a computer or network of computers from unauthorized intruders, which is similar in functionality to the immune system protecting the body from invasion by foreign pathogen. Further, a computer security system should protect against insider attacks, malfunctioning software (analogous to misbehaving cells) and other internal errors, maintaining the computer within normal operating tolerances. Because of the compelling similarity between the computer security problem and the problem of protecting a body against damage from internally and externally generated threats, we designed an artificial immune system to protect computer networks based on immunological principles, algorithms and architecture[8,7].

2.1. Defining self

The security of computer systems depends on such activities as detecting unauthorized use of computer facilities, maintaining the integrity of data files, responding to ``denial-of-service" attacks, and detecting and eliminating computer viruses. We view these protection problems as instances of the more general problem of distinguishing self (legitimate users, uncorrupted data, etc.) from dangerous nonself (unauthorized users, viruses, and other malicious agents). Just as the natural immune system evolved to monitor certain observables in the body (e.g., peptides or heat-shock proteins), so must an artificial immune system be designed to monitor certain aspects of a computer.

We briefly describe an intrusion-detection system built to protect networked computers, although we have studied other problems, including computer virus detection[6] and host-based intrusion detection[5]. Our intrusion-detection system treats self as being synonymous with normal behavior of a local-area network. This dynamical view of self is quite different from that taken by typical anti-virus software, which looks for changes in files stored on hard disks. The distinction is roughly analogous to that between gene products and genes themselves. For example, if the natural immune system had evolved to inspect directly the genomes of all cells for irregularities, we would have a system more closely analogous to anti-viral software. Instead, the immune system typically monitors gene products.

Our equivalent of an organism is a local area network (LAN) of computers. TCP/IP is the most common connected communication protocol used on the internet, and the behavior of our model organism can be characterized by its TCP/IP connections, or *datapath triples*[10]. A datapath triple is comprised of a source address, destination address and communicating program, and this information completely specifies a network connection. Our equivalent of a "peptide" in this environment is a binary string representing the datapath triple. All normally observed and acceptable connections, both those within the LAN and those connecting the outside world to the LAN, form the set of self patterns, and all others (potentially an enormous number), form the set of nonself patterns.

2.2 Architecture

Network traffic is monitored by a set of *detectors* on each computer in the LAN. Each detector consists of a binary string (analogous to a receptor on a lymphocyte), and a detection event is a partial match between a detector string and a datapath string (analogous to the binding between a receptor and an epitope). The partial matching is implemented by a threshold-based rule, for example, two strings match if they have more than a given number of bits in common. Partial-matching can be thought of analogously to the affinity of a receptor to a ligand.

Detectors have a finite lifetime, and follow a lifecycle reminiscent of the lifecycle of immune system cells, such as T-cells and B-cells. Initially, a detector is created with a randomly generated receptor string, and remains immature for a certain period of time. During this maturation period, the detector is compared to all occurring datapath triples (network connections), and if the detector matches any triples, it ``dies" and is replaced by a detector with a new, randomly generated receptor string. This is analogous to negative selection of thymocytes in the thymus. If the detector survives

the maturation period without matching anything, it becomes mature, and future matches raise an alarm, indicating that a potentially dangerous network connection pattern was detected. A mature detector has a finite lifespan, after which it is replaced by a new, immature detector.

Negative selection is an important feature of our system, because it allows for *distributed detection*. Once censored by the negative-selection process, each detector can function independently of other detectors, that is, without communication between detectors or coordination of multiple detection events. This is because each detector covers part of nonself. Thus, a set of detectors can be split up over multiple sites, which will reduce the coverage at any
given site but provide good system-wide coverage. To achieve similar coverage using detectors which match against self would be computationally inefficient.

3.0 Lessons learned

We have described the basis for our artificial immune system. In experiments, this simple start was not sufficient to create an effective security system. We had to add several more immunological mechanisms to address limitations in the basic model. Because these mechanisms were added to solve specific problems, we believe that we can offer some novel perspectives on why these mechanisms might exist in the immune system.

3.1 Tolerance and the reduction of autoimmunity

In our negative-selection algorithm, we assumed that detectors would be exposed to a comprehensive sample of self during their maturation period. This was not always the case, and the problem is exacerbated by legitimate changes to self. In the network, legitimate changes to the profile of self patterns might occur when a computer is added or dropped from the network, when a new user is added to the network, or when new software is installed. These new self patterns caused unacceptably high *false positive* rates analogous to autoimmune reactions.

To reduce the resultant false positives, we implemented a mechanism similar in spirit to the costimulation that a B-cell must receive from a T-helper cell. In our case, the second costimulatory signal was provided by a human observer. If a mature detector does not receive this signal within a given period after matching (typically 24 hours), it will die. Consequently, detectors responsible for false positives are automatically eliminated, whereas a human intervenes to confirm true positives. This allows the system to adapt to incomplete or evolving definitions of self. It also allows for shorter maturation periods in the negative-selection phase, and hence a higher ratio of mature to immature detectors. Costimulation in the natural immune system presumably has similar benefits, as well as protecting against potentially inappropriate somatic mutations.

The form of tolerization we have described is similar to peripheral tolerance in the immune system, in that mature detectors are still subject to tolerization through costimulation. This form of tolerization is essential when the description of self is gradually changing, or when it is impossible to accumulate a complete description of self in a single location. However, it can be very

inefficient. One way to address this inefficiency is to accumulate as many self strings as possible in a single safe location and store them, so that new detectors can be compared against all strings in the store. In this way, detectors can be generated that are more likely free from nonself contamination and they can be generated more efficiently, reducing the need for lengthy maturation periods. This is the equivalent of central tolerization in the thymus, and illustrates that the ideal system may be one in which as much tolerization as possible is carried out centrally, with peripheral tolerization being used to address the issues of perpetual novelty, incomplete descriptions of self, and somatic mutation. There have been debates in immunology about the relative roles of tolerization in the thymus and peripheral tolerization; our experiences suggest that both are essential, because they play complementary roles.

3.2 Finite lifetimes and long-lived memory cells

The perfect detection system would have a single detector for every nonself string, and the detector would match only that nonself string and no other. However, such perfect detection would require as many detectors as there are nonself strings: an infeasible number. The first mechanism for overcoming these resource limits is *generalization*: because of partial matching, each detector can detect a subset of nonself strings. As detectors become more general, a single detector can match a larger subset of nonself, and so fewer detectors are needed. However, as the generality increases, so the ability to make precise discriminations decreases, diminishing the detection abilities of the system. Further, as the probability of an immature detector matching self increases, it takes longer to generate mature detectors that are tolerized to self. Hence, the generality of detectors is limited.

It is likely that a detector set that is limited both in generality and numbers will fail to detect at least some important nonself patterns. To combat this problem, we introduced a second mechanism, *dynamic coverage*: the detectors in the detector sets are continually changing. This dynamic coverage is a consequence of randomly generating detectors coupled with finite lifetimes. Dynamic coverage ensures that an attacker cannot repeatedly exploit the same gaps in coverage. There is a trade-off here, however. As the lifespan of an individual detector decreases, the potential for exploiting a hole in the coverage decreases, but a detector spends relatively more of its life in the maturation phase, where it is not contributing to detection at all. We expect the immune system to be subject to a similar trade-off, one which will govern the optimal lifespan of lymphocytes and other cells.

However, not all of our detectors have a finite lifespan. Those detectors that have detected anomalies and received human-mediated confirmation enter a competition where the best-matching detectors become *memory* detectors. Memory detectors are analogous to long-lived immune memory cells, in that they have much extended lifespans, and have lower thresholds of activation. Memory detectors greatly enhance detection of previously seen attacks by automatically extracting and encoding signatures of attacks. The efficacy of this mechanism reflects the efficacy of secondary responses in the immune system.

3.3 MHC and diversity

Generalization is an important tool in a resource-limited environment: if each detector can match a subset of nonself patterns, fewer detectors are needed. However, generalization introduces potential discrimination errors in the form of *holes*: a hole is a nonself string for which no valid detectors can be generated [3,4]. That is, any possible detector that could match patterns in the hole would also match some patterns in self. As the generality of detectors increases (the specificity decreases), so the potential for holes also increases. Holes are problematic. Once discovered, they can be continuously exploited by an attacker, because holes are a consequence of the structure of the self set, and cannot be overcome by dynamic coverage.

A solution that proved to be effective at reducing the overall number of holes is *multi-representation*---different representations are used for different detectors. One way of achieving this is for each detector to have a randomly generated permutation rule, according to which all datapath triples are permuted before being matched against the detector. This effectively changes the structure of the self set for each detector, with the result that different detectors will be subject to different holes. Consequently, where one detector fails to detect a nonself triple, another may succeed. Multi-representation was particularly effective at reducing the number of holes when the nonself patterns were similar to self patterns.

Similarly to our artificial system, the immune system also faces problems of limited resources, and appears to use both generalization and dynamic coverage. Generalization is a consequence of the fact that a monoclonal lymphocyte can bind to a set of structurally similar peptides, which is analogous to partial matching. It is not unreasonable to assume that this generalized detection also results in holes, and if so, pathogens will evolve away from detection towards the holes. We speculate that molecules of the Major Histocompatibility Complex (MHC) implement a form of multi-representation. Each different type of MHC can be regarded as a different way of representing a protein (depending on which peptides it presents); in effect, the immune system uses multiple representations of proteins. Hence, varying the MHC varies the holes that exist. This is illustrated by the existence of diseases, such as leprosy, that are strongly affected by MHC types. This perspective on MHC can give us insights into the evolution of MHC.

4. Conclusion

In summary, we were surprised at how many features of the natural immune system we were forced to incorporate in order to achieve acceptable performance of our artificial immune system. Studies such as these can help shed light on the question of what role different components and mechanisms play in the natural immune system, and they can provide a partial answer to those who argue against teleological explanations of the immune system. In the long run, understanding what role different components play and why they evolved will help us design more effective and robust interventions and therapies.

5. Acknowledgments

The authors gratefully acknowledge the support of the National Science Foundation (grants IRI-9711199, CDA-9503064, and ANIR-9986555), the Office of Naval Research (grant N00014-99-1-0417), Defense Advanced Projects Agency (grant AGR F30602-00-2-0584) and the Intel Corporation. Many of the ideas described in this paper were developed during many years of collaboration with Alan Perelson and Lee Segel.

6. Bibliography

[1]Burnet and F. Fenner. The Production of Antibodies. London: Macmillan, 2nd edition, (1949).

[2] I.R. Cohen. "Discrimination and dialogue in the immune system." *Immunology*, 12:215--219, (2000).

[3] P. D'haeseleer, S. Forrest, and P. Helman. "An immunological approach to change detection: algorithms, analysis and implications." In *Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy*. IEEE Press (1996).

[4] Patrik D'haeseleer. "An immunological approach to change detection: Theoretical results." In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press (1996).

[5]S. Hofmeyr, A. Somayaji and S. Forrest "Inusion detection using sequences of system calls." *Journal of Computer Security*, vol. 6, pp. 151-180 (1998).

[6] 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 Alamitos, CA. IEEE Computer Society Press (1994).

[7] S. A. Hofmeyr and S. Forrest. "Architecture for an artificial immune system." *Evolutionary Computation Journal*, 8(4):443-473 (2000).

[8]Steven A. Hofmeyr. "An immunological model of distributed detection and its application to computer security." PhD thesis, University of New Mexico, Albuquerque, NM. (1999).

[9] P. Matzinger. "Tolerance, danger, and the extended family." *Annual Reviews in Immunology*, 12:991-1045 (1994).

[10]B. Mukherjee, L. T. Heberlein, and K. N. Levitt. "Network intrusion detection." *IEEE Network*, pages 26-41 (1994).

[11]R. E. Vance. "Cutting edge commentary: A Copernican revolution? Doubts about the danger theory." *Journal of Immunology*, 165:1725-1728 (2000).