Intrusion Detection: Immunologically Inspired Approaches

What can we learn from the human immune system to create adaptable intrusion defense technologies?

BY DEVID PIPA AND ALEXANDER W. DENT

Royal Holloway
University of London

SELF-LEARNING AND ADAPTABLE INTRUSION DEFENSE

We live in an era where information has become one of our primary commodities. This information is in many forms. A large amount of it is kept electronically in means such as USB sticks, smart cards, home computers, large mainframe computers, and networked systems. One of the primary issues to be addressed when talking about the importance of this information is its security.

A large part of information security is taken by the concept of intrusion detection and prevention. Nowadays anyone, ranging from an end-user to a CTO, is familiar with protection methods such as firewalls and anti-virus programs. These are purpose designed pieces of software with the objective of helping to assure the desired level of **confidentiality**, **integrity** and **availability** of information.

The traditional methods of intrusion detection and prevention have a rather widespread use and in terms of their efficiency, they have reached a rather high standard. However, there is one major leak in all of these systems. They are based on a static set of rules and individuate malicious activity through the use of predefined static signatures. So, if a signature for a certain type of attack does not exist within the system, the attack will not be detected and the payload of the attack will not be prevented.

With the large growth in software size and complexity, it has become impossible to predict all possible actions and behavior, and create signatures for all possible malicious conducts. It is quite safe to say therefore, that these security systems need a level of self-learning and adaptability. They must become capable of training themselves to recognize new types of attacks. At first this might seem like an impossible goal, but there are many systems with similar principles in nature that we may use as inspiration. The most obvious of these is the human immune system. This is a system that has been fighting intrusions for a much longer time and has continuously been adapting itself to ensure our survival. Considering the latest estimates on the world's population it's safe to say that it has been doing a rather good job and, given its experience, there is probably a thing or two to learn from it.

Devid Pipa
Information Security Group
Department of Mathematics
Royal Holloway University of London
devidpipa@gmail.com

Alexander W. Dent
Information Security Group
Department of Mathematics
Royal Holloway University of London
a.dent@rhul.ac.uk

This article was prepared by students and staff involved with the award-winning M.Sc. in Information Security offered by the Information Security Group at Royal Holloway, University of London. The student was judged to have produced an outstanding M.Sc. thesis on a business-related topic. The full thesis is available as a technical report on the Royal Holloway website http://www.ma.rhul.ac.uk/tech.

For more information about the Information Security Group at Royal Holloway or on the M.Sc. in Information Security, please visit http://www.isg.rhul.ac.uk.

Intrusion Detection: Immunologically Inspired Approaches

1. THE HUMAN IMMUNE SYSTEM

Human immunology was established as a discipline of its own in medicine in the late 1950s with the discovery of the different molecules, cells and organs of the human body.

This science distinguishes two types of immune defenses present within our bodies:

- Innate immunity
- Adaptive immunity

Figure 1: www.halonsecurity.co.th

Large amounts of time and resources are invested in trying to find system vulnerabilities and patching them. Wouldn't the dream scenario be the one where the system does all of this on its own? All we would have to do is sit back, relax and watch hackers waste their time.

Innate immunity is the set of immunological defenses that we are born with and is static throughout our lifetime. Adaptive immunity is the set of immunological defenses that an organism develops throughout its life. Both these types of defense protect our bodies from the threat of pathogenic substances. Pathogenic substances are any type of cellular based micro-organism that, if introduced into our system, would create a deterioration of our health.

The innate immune system is a collection of defenses that are passed onto an individual from his or her parents. Therefore, it is unable to detect a pathogenic substance that has not been previously met by the ancestors of the individual in question. This is quite similar to the current methods of intrusion detection in information security. For example, one of the levels that innate immunity operates at is the skin itself. This organ has the function of preventing unwanted agents from penetrating the body. It only allows substances to go through it if certain size and shape criteria are met. This is similar to methods of intrusion prevention like firewalls.

Adaptive immunity, as the name suggests, is the set of immune defenses that a body gains throughout its lifetime. The way this

The innate immune system is a collection of defenses that are passed onto an individual from his or her parents.

system works is by creating a large set of defenses. Each entry in this set, corresponding to a particular defense, is able to recognize only a group of similar pathogenic substances. This property is called high-specificity. Together, all the different entries in the set aim to cover all possible pathogenic substances. However, in order to generate a defense towards a new pathogenic substance, there must be a contact between the substance in question and the system.

The system makes use of a set of cells called lymphocytes. These flow through the bloodstream and monitor all other cells present in the body and perform a process named self/non-self discrimination. This is the process by which cells that do not belong to our bodies are discriminated against the ones that do belong. Lymphocytes perform such actions through the monitoring of three dimensional protein structures that are present on the surface of any organic cell, called epitopes.

Lymphocytes present on their surface a set of structures called receptors, which are able to bind to non-self protein structures if they are complementary. This level of complementarity is called affinity. If a level of affinity is met then, a bind is established and the cell is classified as non-self. It is worth mentioning at this point that lymphocytes are generated in our bodies in such a way to only be able to bind to non-self structures. This is done through a process named negative selection. A visual example of a lymphocyte and its affinity towards different pathogenic substances if shown in Figure 2.

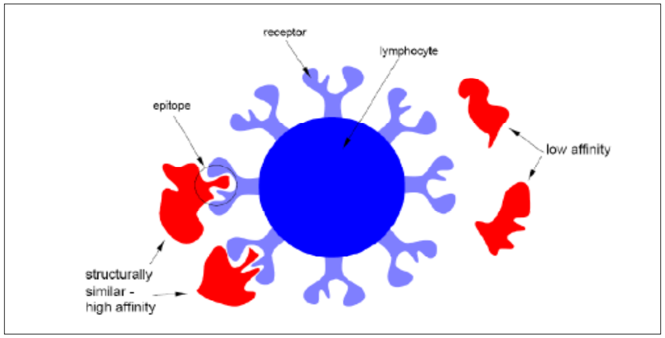

Figure 2: A visual representation of a lymphocyte and its affinity to a set of pathogenic substances. Receptors on the same lymphocyte are identical and they are high specific to certain types of pathogenic substances. This is the affinity level. Extracted from [Hof99].

There are a number of properties of the adaptive immunity that are of interest to this study:

1. **Self/non-self discrimination:** The way in which the human immune system is able to distinguish between cells that belong to the body and cells that have intruded into

Lymphocytes present on their surface a set of structures called receptors, which are able to bind to non-self protein structures if they are complementary.

the system with malicious intentions. This decision is made according to the affinity level.

2. **Negative selection:** The initial creation of lymphocytes is an entirely random process that happens in the bone marrow. This means that some lymphocytes might present receptors that would bind to self structures. Negative selection ensures that these lymphocytes do not enter the bloodstream. After creation lymphocytes migrate to the thymus and the ones that are complementary to self structures are eliminated. This is a very important part of the immune system. Human immunity has the power to entirely annihilate our bodies. Negative selection is the process that ensures this does not happen.

3. **Dual authentication:** Lymphocytes are subdivided into T-Cells and B-Cells. The B-cells are responsible for the recognition of the non-self structures. These then need the authorization of T-cells to proceed to the elimination phase. T-cells are the ones that make sure that a cell classified as non-self by a B-cell is indeed non-self and not a misindentification.

4. **Hypermutation:** This is one of the most interesting properties of the immune system and contributes towards its adaptability and helps it be ever-changing. Once a bind is established between a B-cell lymphocyte and a non-self structure, and this bind is authenticated by a T-cell, that particular lymphocyte is cloned and its receptors undergo a hypermutation process through which they are made more and more complementary to the recognized pathogen. The clones also undergo the negative selection process of course. This process helps increase the level of specificity of lymphocytes. Upon a future encounter with the same or similar pathogenic substance, due to the hypermutation process, the body will have better defenses available in a much shorter time as no resources will have to be wasted in adapting towards the pathogenic substance in question.

The body does not maintain a full set of lymphocytes to cover all possible non-self structures that could be encountered. This would be a huge waste of resources. When a pathogenic substance is encountered, the body builds the necessary defenses through adaptability. This does not mean that lymphocytes for all previously encountered pathogens will always be present. The

The body does not maintain a full set of lymphocytes to cover all possible non-self structures that could be encountered.

system instead maintains a memory of how to build these, so upon future encounter, the adaptation phase can be skipped and the defenses can be rebuild much quicker.

2. BUILDING A COMPUTER IMMUNE SYSTEM – ARCHITECTURES

The human immune system is able to adapt itself towards the recognition and destruction of new pathogenic substances. Is it possible to adapt the knowledge gained from the human immune system into a security system for computers? There is no straightforward yes or no answer to this question, so let us take a closer look at the possibilities.

The first thing to consider would be the framework architecture for such a system. In the 1997 paper entitled “Principles of a computer immune system” by Anil Somayaji, four possible architectures are presented:

1. **Protecting static data:** This architecture takes the approach of combating malicious behavior through the monitoring of static data on the hard drive. Self is defined as the normal set of instructions of the programs. This architecture is not deemed to be the most favorable one though. First of all, a large amount of time is required to analyze data on hard drives due to lack of speed, and secondly, any possible corruption will not have any negative consequence on the system until the program is run.

2. **Protecting active processes on a single host:** Consider a scenario where the analogy between a computer and a human body is as follows:

- every active process in a computer is considered as a cell
- a computer running multiple processes would represent a multicellular organism
- a set of computers would represent a population of these organisms

In this case, the intrusion detection could be done through the implementation of a lymphocyte process that is able to monitor the running of other processes. Self would be defined as the normal behavior of a process and non-self would be classified as any behavior that does not fall within normal bounds. The “lymphocyte” process would have the power to kill, restart or halt a process that is acting not in accordance with the definition of normal behavior.

3. **Protecting a network of mutually trusting computers:** This architecture poses a view where a network of computers

The “lymphocyte” process would have the power to kill, restart or halt a process that is acting not in accordance with the definition of normal behavior.

is seen as a human body. Each computer on the network is an organ of the body. All the other principles and methods of work of the previous architecture apply in the same manner; however, in this model the computers have a level of trust between each other and lymphocyte processes running on these machines would be able to migrate between them.

The clear disadvantage of this approach however is that if a vulnerability is exploited on one machine, and none of the others "pick up on it", the entire set of linked machines becomes vulnerable.

4. **Protecting a network of mutually trusting disposable computers:** In this architecture the network is divided into two "subnets" which are able to communicate between each other. One subnet consists of the normal set of computers for the users. The second subnet consists of a number of computers that are carrying out the lymphocyte task. These are in charge of monitoring all activity on the other computers and the activity between themselves too. If anomalous behavior is detected on any machine they have the power to shut down, restart or maybe halt that machine or the anomalous process on that machine. This would mean that the system would have to be able to cope with "self sacrificing" machines and that the function of the system must still be able to be completed even if some of the machines on that system are lost.

3. BUILDING A COMPUTER IMMUNE SYSTEM – METHODS & ALGORITHMS

After presenting possible architectures, the main issue to be dealt with is: "How do we define self in a computer system?" In the human immune system this is empirically defined in the DNA. In computers, only one realistic method has been proposed for the assessing whether a process is self (normal) or non-self (malicious). The "lymphocytes" in the system monitor the sequence of system calls generated during the execution of a process. Of course, every execution of a process will generate a different sequence of system calls; however, it is believed that small subsequences within these large sequences will always be present and relatively similar.

Data is collected using a windowing method. A window of preset size is slid across the sequence by one position at a time. For every window and each system call

After presenting possible architectures, the main issue to be dealt with is: "How do we define self in a computer system?"

in the window an entry is created and all following calls and their positions are recorded into the database. Each window will then define a subsequence, therefore an entry into the database.

Having created the self database, the next step to take would be to generate detectors acting as the receptors on the lymphocyte. There is however one issue to be tackled before moving to this step. The lymphocyte receptors perform the self/non-self discrimination according to the affinity level. How do we define affinity in our scenario? In current research, the most popular way of doing this is through string matching. So, supposing there is a non-self database against which we are testing, strings of system calls of preset size will be matched against this. There is a number of ways of performing string matching, such as the r-contiguous bits or Hamming distance. For ease of example let us use the r-contiguous bits rule over two strings where each letter represents a system call.

MATCH	NO MATCH
abacdabdcabacbdc	abacdabdcabacbdc
acdbdabdcababadc	abdcbadbcabdaaab

So, for string size 16 and r = 8, the two strings are said to be a match if identical substrings of 8 bits can be found in both of them. The downside of this approach is that only the system calls are being monitored, and also the data passed by each one of these calls is being left out.

Having established the matching rule, representing “affinity”, the next step is to generate the non-self database. The principle of negative selection must be recalled at this point. With compliance to the matching rule we must generate a database of strings that represent out of normal, non-self, behavior. The obvious way to generate candidate strings that represent non-self behavior is to generate them completely at random and then delete them if they match to any strings in the database of self strings. This has the disadvantage of potentially taking a long time to generate a string that doesn't match the system. The system might also generate a set of strings which are very similar, meaning that there are lots of deviant processes which wouldn't be caught by the system at this time. Luckily, there are better string generation algorithms which can build strings faster and in such a way that more sequences representing behavior that is

With compliance to the matching rule we must generate a database of strings that represent out of normal, non-self, behavior.

deemed as deviant can be caught.

Of course, the larger the number of strings in the database, and the longer these strings are, the more likely it is that any non-self behavior will be detected. However, the more strings that are in the database, and the longer these strings are, the more time it will take to check for a match. So, the size of the database and the size of the strings themselves can be viewed as determining a trade-off between security and efficiency. The larger the database, the more secure the system will be, but the slower it will run.

Once the non-self database is generated, the system is ready for deployment. During execution, the self/non-self discrimination will be performed through comparing subsequences of the system calls sequence for any given process against the database for that process. The windowing method used for the creation of the self database will be used for the self/non-self discrimination too. The sequence of system calls generated by the running of a process is subdivided into substrings using this method and each window is then compared to each entry in the non-self database.

4. BUILDING A COMPUTER IMMUNE SYSTEM – SOME ISSUES

A number of methods and algorithms that analogize the human immune system have been created for use in the scope of computer security; however some words have to be said about how good they would be if to be used in the everyday environment.

First of all we have the definition of self that is gained in a training process through the monitoring of normal system usage. Looking at the human immune system, this is not how this is done. Self definition is there right from the start. It is defined in our DNA. Our entire organism starts from just one cell. Given that cell, where any possible acceptable variation of self is defined in the DNA string, others are created. Would this not be more similar to defining self in a computer program at function level in the source code?

Secondly, the way in which the self/non-self discrimination is performed is done in a simple one dimensional string matching process. In the human immune system, this is done through the monitoring of much more complex structures in a greater number of dimensions.

Looking at the entire picture from a broader view also, we are trying to create a security

The larger the database, the more secure the system will be, but the slower it will run.

system based on the principles and methods of work of another security system. However, it is worth mentioning that these two systems have entirely different objectives. The human immune system has the objective of keeping the organism alive. In this prospect, if some cells are killed by a pathogenic substance it is not a problem. The body will not die because of this and the cells will be regenerated. In a network of computers on the other hand, we can not afford to lose any of them. The consequences of this might be catastrophic (even supposing that the system is successful at 100% and detects all attacks). If a process is killed or restarted, the data contained in that process would be lost. Not much of an improvement from the previous scenario.

However, there are indeed cases where it may be affordable to lose some computers and still not have the system compromised: the case of computer farms for example. A computer farm is a large system of computers that share heavy calculations. If one or more of them are compromised, this will not be a problem for the rest of the system, as, temporarily, the work of these can be carried out for the others.

It has to be noted however that adaptive immunity systems should not be the only defense that a system uses. Instead they should be run as an extra layer on top of the already existing and traditional methods. This is analogous to human defenses, which are a combination of innate immunity and adaptive immunity systems. For example, given the current process for the creation of a non-self complementary database, a buffer overflow would not be detected by such a system, as the series of system calls would be the same. However, it is possible that if the buffer overflow attack is used for the injection of malicious instructions into the code of a program, these may be detected later and their execution will be prevented.

Some words must be said about the time that such an adaptive immune system would require to run. It would take a long time to compare each string of system calls to each entry in the non-self database at least. Also, during this time, a potential attack would go undetected and would be free to deliver its payload until detected. Concerning the time complexity mentioned before, this would be increased many times in a networked environment, where everything would be matched against all computers.

If a process is killed or restarted, the data contained in that process would be lost.

5. BUILDING A COMPUTER IMMUNE SYSTEM – SOME IDEAS

Two of the principles of the human immune system remain untouched to date: hypermutation and dual-authentication. These are both very important components of the immune system. Hypermutation guarantees its effectiveness in the long run term and dual-authentication makes sure of the efficiency of this system in terms of false positives.

Previously it was mentioned that due to the time needed to individuate an attack, the attack would have more time to deliver its payload. A hypermutation process could help towards the reduction of this time. Currently systems treat each infection individually, even if the system has experienced the same attack before.

Dual authentication would also be an important point to look into. A large number of false positives might arise in such a system due to lack of efficiency in the training process for the generation of the self database. User behavior varies greatly between individuals. If the database is created through the behavior of only one user, then the non-self detectors generated by basing on this database, would most likely include normal behavior of a different user. ⁕

Ron Condon

UK bureau chief
searchsecurity.co.UK

Ron Condon has been writing about developments in the IT industry for more than 30 years. In that time, he has charted the evolution from big mainframes, to minicomputers and PCs in the 1980s, and the rise of the Internet over the last decade or so. In recent years he has specialized in information security. He has edited daily, weekly and monthly publications, and has written for national and regional newspapers, in Europe and the U.S.

REFERENCES

[Hof99] – "An Immunological Model of Distributed Detection and its Application to Computer Security" Steven Andrew Hofmeyr – May 1999.