APPLYING SIMILARITIES BETWEEN IMMUNE SYSTEMS AND MOBILE AGENT SYSTEMS IN INTRUSION DETECTION

Marek Zielinski, Lucas Venter

School of Computing, University of South Africa

Marek Zielinski (contact author):
E-mail: marekz@webmail.co.za
Telephone: +27 (0)73 427 6167
Postal Address: P O Box 15282, Secunda, 2302, South Africa
Lucas Venter:
E-mail: ventelm@unisa.ac.za
Telephone: +27 (0)12 429 6368
Postal Address: P O Box 392, UNISA, 0003, South Africa

ABSTRACT

Nearly all present-day commercial intrusion detection systems are based on a hierarchical architecture. Nodes at the bottom of the hierarchy collect information, which is passed to higher nodes in the hierarchy until the root node is reached. The root node is a command and control system that evaluates attack signatures and issues responses. Many single points of failure exist in an intrusion detection system (IDS) based on a hierarchical architecture that does not have redundant communication lines and the capability to dynamically reconfigure relationships in the case of failure of key components. For example, an attacker can cut off a control branch of the IDS by attacking an internal node or even interrupt the operation of the entire system by taking out the root command and control node.

To solve this problem, we propose an IDS inspired by the human immune system. The architecture of the proposed IDS has no aggregation nodes or a root node that evaluates attack signatures. Instead, the function of attack signature evaluation is divided and placed within mobile agents. The mobile agents act similarly to white blood cells of the immune system and travel from host to host in the network to detect any intrusions. As in the immune system, intrusions are detected by distinguishing between "self" and "non-self", or normal and abnormal process behaviour respectively. The IDS can remain operational even when most of its components have been disabled because the agents that remain in the network can still carry out their task as they do not need to communicate with their home platform. Furthermore, because mobile agents are not static and their number can vary, the whole IDS is more difficult to disable than an IDS based only on static components.

KEY WORDS

intrusion detection, immune system, mobile agent, computer security

APPLYING SIMILARITIES BETWEEN IMMUNE SYSTEMS AND MOBILE AGENT SYSTEMS IN INTRUSION DETECTION

1 INTRODUCTION

A computer system's security mechanisms should prevent unauthorized access to its resources and data. However, it is impossible to build a completely secure system for many different reasons: programs and operating systems have vulnerabilities, firewalls can be circumvented, passwords can be cracked, and a system can be abused by insiders (Sundaram, 1996). Moreover, the increasing connectivity of computer systems gives greater access to outsiders and makes it easier for intruders to avoid identification (Mukherjee, Heberlein & Levitt, 1994).

Unless there is a mechanism to detect breaches of the system's security, we may be unaware that the computer system has been attacked. *Intrusion detection* provides such a mechanism and is defined as "the problem of identifying individuals who are using a computer system without authorization (i.e. 'crackers') and those who have legitimate access to the system but are abusing their privileges (i.e. the 'insider threat')" (Mukherjee, Heberlein & Levitt, 1994).

Nearly all present-day commercial intrusion detection systems (IDSs) follow a hierarchical architecture (Jansen, 2002). Hierarchical architectures are used because they are excellent for creating scalable distributed IDSs with central points of administration. However, an IDS based on a hierarchical architecture has many single points of failure. For example, by disabling the root node, which evaluates attack signatures, the whole IDSs will be disabled (see section 2.2).

To solve this problem, we propose an IDS inspired by the human immune system. The IDS uses mobile agents which travel from host to host in the network to detect intrusions. By using mobile agents, the system is made more resistant to failure because it can remain operational even when most of its components have been disabled.

The rest of this paper is organized as follows. Section 2 briefly discusses intrusion detection systems and the problems associated with a hierarchical architecture. Mobile software agents and their benefits for intrusion detection systems are discussed in Section 3. In Section 4, a brief overview of the human immune system is presented together with the properties that enable it to effectively detect intrusions. Section 5 discusses the main aspects of how an IDS based on the human immune system and mobile agents could be constructed as well as how this IDS provides greater fault-tolerance. Section 6 discusses how the concepts of the immune system have been applied in the IDS. Conclusions and future work are presented in Section 7.

2 INTRUSION DETECTION SYSTEMS

An *intrusion detection system* (IDS) is an automated system that aims to detect intrusions in a computer system. The main goal of an IDS is to detect any unauthorized use, abuse, or misuse of computer systems by both system insiders and external attackers (Mukherjee, Heberlein & Levitt, 1994). Its purpose can be compared to that of a car alarm, which alerts its owner when the car has been broken into.

2.1 Classification of intrusion detection systems

Intrusion detection systems can be classified according to:

1. The source of data used for analysis
2. The intrusion detection model used by the IDS
3. The distribution of the IDS components

With respect to the source of data used for analysis, intrusion detection systems are classified as host-based or network-based (Mukherjee, Heberlein & Levitt, 1994). Host-based systems operate at the host level, using the operating system's audit trails as the main source of data input to detect intrusive activity. Network-based systems use raw network packets as the data source. The network packets are obtained from a network adapter card running in promiscuous mode (a mode during which all the frames that pass over the network are picked up; not just those destined for the node served by the card). Early IDSs were host-based, but present systems are usually network-based. Although most of the network-based IDSs build their detection mechanism on monitored network traffic, some also use host audit trails.

The intrusion detection model used by the IDS refers to what techniques are used by an IDS to detect intrusions. IDSs mainly use two techniques: anomaly detection and misuse detection (Bace & Mell, 2001). It is also possible to use a combination of both anomaly and misuse detection techniques (Botha, 2004).

An IDS that employs anomaly detection first creates a profile of normal system behaviour, during which no intrusion takes place. Once the profile has been created, the IDS compares the current behaviour of the system with the behaviour recorded earlier. It is assumed that any intrusive actions will result in behaviour different from that normally seen in the system. Any significant deviations from the normal behaviour are treated as intrusions. Although an IDS that uses anomaly detection can detect new types of attacks, it has a high false positive (false alarm) rate.

An IDS that employs misuse detection is based on searching for known attacks in the behaviour of the system and its users. The advantage of this approach is that known attacks can be detected accurately, which leads to low false alarm rates. However, previously unseen attacks cannot be detected.

With respect to component distribution, intrusion detection systems are classified according to the way in which their components are distributed. Spafford and Zamboni (2000) identify two such classes: centralized and distributed intrusion detection systems. A centralized IDS analyses its data at a fixed number of locations. These locations are independent of the number of hosts being monitored. A distributed IDS analyses its data at a number of locations proportional to the number of hosts that are being monitored. Only the locations and the number of the data analysis components are considered in this classification; the data collection components are not considered.

2.2 Hierarchical IDSs

Nearly all present-day commercial IDSs follow a hierarchical architecture (Jansen, 2002). In this section, we briefly discuss this type of architecture by using the work of Jansen (2002), Jansen et al. (2000) and Jansen et al. (1999). We will draw from these sources here without any further reference to them.

A hierarchical architecture follows a tree structure as shown in Figure 1. Individual nodes within a network are shown with circles and the information flows between different types of nodes are shown with arrows.

The leaf nodes represent network-based or host-based collection points at which information is gathered. The event information is passed to internal nodes, which aggregate information from multiple leaf nodes. Further aggregation, abstraction and data reduction occurs at higher internal nodes until the root node is reached. The root node is a command and control system that evaluates attack signatures and issues responses. Typically, the root node also reports to an operator console where an administrator can manually assess status and issue commands.

Reliance on hierarchical structures for components makes the IDS vulnerable to direct attack. Many single points of failure exist in an IDS that has no redundant communication lines or the capability to dynamically reconfigure relationships in the case of failure of key components. For example, an attacker may interrupt the operation of the entire IDS by successfully disabling the root node. Since the root node is responsible for evaluating attack signatures and for issuing responses, disabling it will not allow the IDS to detect intrusions. The critical role played by this central controller makes it a likely target of attack. Although such critical components usually reside on platforms that have been hardened to resist direct attack, the IDS may still be vulnerable as other survivability techniques such as redundancy, mobility, or dynamic recovery are lacking in current implementations. A system could also employ redundant components for each key node to avoid this problem. However, such a solution does not offer much fault tolerance because a determined and knowledgeable attacker can disable a small number of backups.

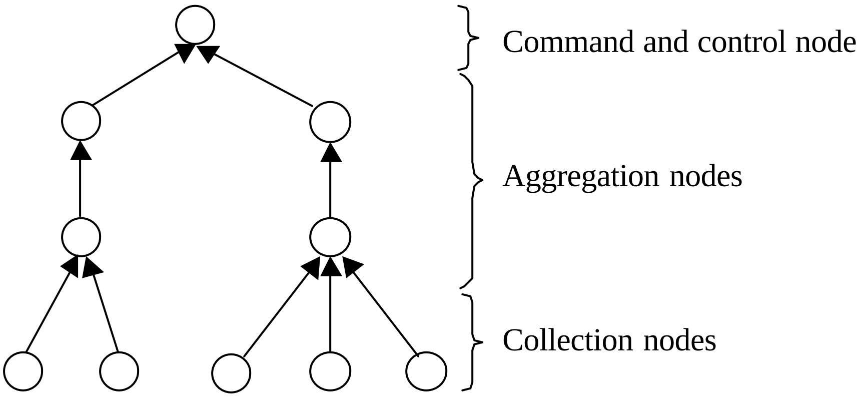

Figure 1: Hierarchical architecture of an IDS

3 MOBILE SOFTWARE AGENTS

3.1 An overview of mobile agents

A *software agent* can be defined as (Bradshaw, 1997) "... a software entity which functions continuously and autonomously in a particular environment ... able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment ... Ideally, an agent that functions continuously ... would be able to learn from its experience. In addition, we expect an agent that inhabits an environment with other agents and processes to be able to communicate and co-operate with them, and perhaps move from place to place in doing so."

Agents execute on agent platforms, implemented on servers and user computers. A platform is usually entirely located at a single node of the network, and multiple platforms may be implemented on a given node. The platform provides a computational environment for the agent to operate, and provides certain services that agents can make use of (Rothermel & Schwehm, 1998).

Agents may be static or mobile. Stationary agents remain resident on a single platform, while mobile agents have the capability to migrate between different platforms. A mobile agent migrates by executing an instruction specifically created for this purpose. For example, mobile agents developed using Telescript (an object-oriented language specifically designed for mobile agent programming) execute the `go` instruction when they decide to migrate (White, 1996). Before a mobile agent migrates, its execution is suspended on its current platform and its state information is saved. During agent migration, both the code and the state information of the agent are transferred to the destination. At the new platform, the mobile agent resumes its execution.

We should note that agent migration is not the same as process migration. The main difference between the two concepts lies in what entity decides when migration takes place and to which destination node in the network. In the case of process migration, the operating system makes this decision and migration is transparent to the process. In the case of agent migration, the agent itself decides when and where to migrate.

3.2 Benefits of mobile agents for intrusion detection systems

There are many benefits derived from applying mobile agents for intrusion detection. Only the benefits relevant to the proposed IDS are discussed in this section by using the work of Jansen (2002), Jansen et al. (2000) and Jansen et al. (1999).

3.2.1 Autonomous and asynchronous execution

IDSs based on mobile agents can continue to operate in the event of failure of a central controller or a communication link. Unlike message passing routines or remote procedure calls, once a mobile agent is launched from a home platform, it can continue to operate autonomously even if the host platform from where it was launched is no longer available or connected to the network. Since mobile agents do not require control by another process, the co-ordination of IDS sensors and filters can be protected from the loss of network connections. Furthermore, a mobile agent would not be prevented from carrying out its assigned tasks if it were unable to communicate with a central controller. Therefore, a mobile agent created to detect intrusions would still be able to do so, even if its home platform is down or unreachable.

3.2.2 Dynamic adaptation

The ability for mobile agent systems to sense their environment and react to changes is useful in intrusion detection. Agents may move elsewhere to gain better position or avoid danger. They can also clone themselves for redundancy and parallelism, or call other agents for assistance. Agents can also adjust to favourable situations as well as unfavourable ones. When combined with autonomous and asynchronous execution, these characteristics facilitate the building of robust and fault-tolerant systems.

3.2.3 Robust and fault-tolerant behaviour

The ability of mobile agents to react dynamically to unfavourable situations and events makes it easier to build robust distributed systems. Their support for disconnected operation and distributed design paradigms eliminate single points of failure problems and allow mobile agents to offer fault-tolerant characteristics.

From this discussion, we can make the following conclusions (Jansen, 2002): Mobile agents have the ability to relocate when sensing danger or suspicious activity, and clone for redundancy or replacement. They can also operate autonomously and asynchronously from where created, collaborate and share knowledge, and be self-organizing (e.g. dynamically reconfiguring relationships to compensate for failure of key components). Therefore, mobile agents can facilitate the implementation of robust and attack-resistant IDS architectures.

4 THE HUMAN IMMUNE SYSTEM

4.1 An overview of the human immune system

The overview presented in this section is largely incomplete as only enough information is provided to understand how its concepts can be applied to an IDS. This overview is based on those presented by Forrest, Hofmeyr and Somayaji (1997); Somayaji, Hofmeyr and Forrest (1997); and Kim and Bentley (1999).

The human immune system is based on the concept of distinguishing molecules and cells of the body, called "self", from foreign ones, called "non-self", and elimination of the latter. The foreign cells are called antigens and include any invaders, such as bacteria and viruses.

The function of the immune system is implemented through the interactions between a large number of different types of cells rather than by one particular organ. The system has multiple levels of defences, from the skin (which forms the outermost barrier of protection) to the adaptive immune system. The adaptive immune system can be viewed as a distributed detection system in the body. The organs of the adaptive immune system, called lymphoid organs, are positioned throughout the body and store lymphocytes, also known as white blood cells.

White blood cells function as small, disposable and independent intrusion detectors that circulate through the body in the blood and lymph systems. Each white blood cell is specialized to ignore self-cells and bind to a small number of structurally related non-self cells. Recognition and binding to non-self cells is accomplished through special receptors on white blood cells. The receptors are structured such that they will bind to a particular peptide (a sequence of amino acids, which make up proteins). As different types of cells contain different proteins, the receptors allow a white blood cell to recognize specific non-self cells and bind to them. After binding, many events still take place, usually resulting in scavenger cells (macrophages) eliminating the antigen.

4.2 Properties that enable the immune system to effectively combat intrusions

Based on a study of the human immune system, Somayaji, Hofmeyr and Forrest (1997), and Forrest, Hofmeyr and Somayaji (1997) have presented many of its properties that can serve as design principles of a computer immune system. The natural immune system has evolved many important properties that enable it to effectively combat intrusions. The properties relevant to the proposed IDS are discussed below.

1. *Distributability*. The immune system is highly distributed. No central co-ordination takes place; infections can be locally recognized by lymphocytes. Distributability greatly enhances the system's robustness.

2. *Diversity*. The immune system of each individual in a population is unique. This ensures that not all individuals will be vulnerable to the same pathogen to the same degree, thereby enhancing the survival of the population as a whole.

3. *Disposability*. Any cell of the immune system can be replaced. Therefore, there is no single component that is essential to the system's function.

4. *Autonomy*. There is no need for outside management or maintenance in the immune system; pathogens are autonomously classified and eliminated.

5. *Adaptability*. The system can adapt by learning to detect new pathogens. At the same time, it is also able to recognize previously seen pathogens through immune memory.

6. *Anomaly detection*. The immune system is said to perform anomaly detection because it is able to detect pathogens that it has not encountered before.

7. *Dynamically changing coverage*. The immune system is unable to maintain a set of detectors large enough to cover the space of all pathogens. Therefore, at any time, only a random sample of detectors circulates throughout the body. This sample of detectors is constantly changing through cell death and reproduction.

8. *Identity via behaviour*. Peptides, or protein fragments, serve as indicators of behaviour through which identity is verified.

9. *Detection is imperfect*. Not every pathogen is matched exactly by a pre-existing detector. This increases the system's flexibility with which it can allocate resources. For example, although a less specific lymphocyte will be less efficient at detecting a particular pathogen, it can detect a greater variety of pathogens.

These properties not only enable the natural immune system to effectively detect and eliminate intrusions, but they also make the system fault-tolerant. The system can remain functional even when many of its components have failed. An IDS design that applies these properties, together with the capabilities and advantages of mobile agents, also increases its fault-tolerance. It too can remain operational even when most of its components have been attacked and disabled. A possible IDS design based on mobile agents and which applies these properties of the immune system is described in the next section.

5 AN IDS BASED ON IMMUNE SYSTEM CONCEPTS AND MOBILE AGENTS

5.1 Overview of the proposed IDS

Somayaji, Hofmeyr and Forrest (1997) have proposed several approaches to building computer security architectures that incorporate principles of the human immune system. The approach most applicable to the suggested solution is to implement the adaptive immune system layer by kernel-assisted lymphocyte processes that can migrate between computers, making them mobile agents. With help from the kernel, the lymphocyte processes are able to query other processes to determine if they are functioning normally. A computer is selected to create and propagate lymphocytes, each of which searches for anomalies in the behaviour of a specific program.

5.1.1 The source of data used for analysis

The IDS uses sequences of system calls of privileged processes as the source of data used for analysis. Therefore, the IDS is host-based because it monitors programs that are executing on different hosts in the network, rather than monitoring the network traffic.

We have decided to concentrate on privileged processes for several reasons. Privileged processes are allowed to bypass the kernel's security mechanism in order to accomplish their tasks. They are also trusted not to compromise the security of the system. However, due to possible errors, privileged programs may have vulnerabilities, which can be exploited by attackers. Therefore, privileged processes are considered more dangerous than user processes because they have greater access to the computer system. In addition, a natural boundary with respect to external probes and intrusions is created by root processes, especially those that listen to a particular port. We have also not decided to monitor user behaviour because the "normal" behaviour of processes is far more limited and stable than the "normal" behaviour of users (Forrest et al., 1996; Ko, Fink and Levitt, 1994). This makes defining normal behaviour much easier and could result in fewer false positives.

5.1.2 The intrusion detection model used by the IDS

In the human immune system, intrusions (in the form of pathogens) are detected by searching for abnormal or "non-self" peptide patterns. The proposed IDS employs anomaly detection, just as the immune system does, because intrusions are detected by searching for deviations from the normal behaviour of programs.

The first step of anomaly detection is to learn what is the normal behaviour of each specific program. The definition of normal behaviour of processes is based on the work of Forrest, Hofmeyr and Somayaji (1997), and Forrest et al. (1996). In their approach, the "peptide" used for recognition of non-self is defined in terms of sequences of system calls executed by privileged processes in a networked operating system. Preliminary experiments performed by Forrest, Hofmeyr and Somayaji (1997) on a limited set of intrusions and other anomalous behaviour show that short sequences of system calls (e.g. of length 6) provide a compact signature for self that distinguishes normal from abnormal behaviour.

Certain aspects of process behaviour are ignored by this definition of normal behaviour, such as parameter values passed to system calls or instruction sequences between system calls. At this stage, this definition is sufficient for our purpose, as our main goal is to demonstrate how an IDS can be designed using mobile agents and certain aspects of the immune system. If, in future, it will be necessary to include other aspects of process behaviour, then only the definition of normal behaviour will need to change.

A separate database of normal behaviour for each process of interest is created by examining the sequences of system calls made during a program's normal execution. The database is specific to a particular architecture, software version and configuration, local administrative policies, and usage patterns. Since there is a large variability in how individual systems are currently configured and used, the individual databases provide a unique definition of self for most systems (Forrest et al., 1996). Several hosts in the network, which will be called lymphoid hosts, store all the different databases defining self.

Once the normal behaviour of a program is known, we can use it to monitor its behaviour whenever it runs. Monitoring is based on examining the behaviour (or the generated sequence of system calls) of an executing program and comparing it with the normal behaviour which has been learnt earlier.

5.1.3 The distribution of the IDS components

Mobile agents analyse data at the same hosts where data is collected. The number and the locations of the data analysis components are dependent on the current number and the locations of the mobile agents in the computer network. Therefore, the IDS is distributed because the number of the data analysis components is not fixed.

5.1.4 The use of mobile agents in the IDS

For mobile agents to be applied to intrusion detection, all the participating nodes must have an agent platform installed. Since many agent systems operate over a wide range of hardware and software, this requirement is not as difficult to fulfil as it may first appear (Jansen, 2002).

There will be one type of mobile agent for each type of privileged program we wish to monitor. At any time, the types of mobile agents that are present in the network determine the types of privileged programs that are monitored. The desired rate of intrusion detection (i.e. how quickly we wish to detect an intrusion) is determined by how many mobile agents of each type are present in the network.

All lymphoid hosts will have the capability to create mobile agents for intrusion detection. However, at any time only one lymphoid host will be assigned the responsibility to create mobile agents, while the other lymphoid hosts will serve as back-ups. When a mobile agent will be created, it will obtain the learned normal behaviour of a particular program (as decided by the lymphoid host) from the normal profile database found on the lymphoid host. It will then travel from computer to computer in the network. At each computer, it will determine if the particular program is running. If it is, then the mobile agent will examine the system calls generated by that running program and compare them with the system calls stored by the agent. If this examination will show that there is a high deviation from the normal behaviour, then the agent will launch an intrusion alert.

When a possible intrusion has been detected, the mobile agent sends an intrusion alert to the lymphoid host, which also provides a user interface through which the system administrator can learn of intrusions and issue commands to the IDS. At this stage, the IDS may decide to increase the rate of intrusion detection for this particular program. This can be done in two ways, depending on how the IDS has been configured. One way is to allow the lymphoid host to create and send new mobile agents, whose type is the same as that of the alerting agent. A second way is to allow the alerting agent to multiply. The number of agent copies that can be created by either way is user-defined. The IDS can also decide to increase the number of different process types that are monitored when an alert is raised. This can be done by increasing the number of agent types that will monitor specific programs.

5.2 How does this IDS provide greater fault-tolerance than the hierarchical architecture?

We cannot design an IDS that will be completely resistant to direct attack. However, we can design an IDS such that it will be more robust and fault-tolerant and thereby more difficult to disable completely. An increased robustness and fault-tolerance of the system will reduce the system's vulnerability to direct attack.

The architecture of the proposed IDS has no aggregation nodes. There is also no root node that evaluates attack signatures. Instead, the function of attack signature evaluation is divided and placed within mobile agents, which travel from host to host in the network.

As explained in Section 4.1, the organs of the adaptive immune system (lymphoid organs) are positioned throughout the body and store white blood cells. Similarly, in the proposed IDS, the databases that define self are distributed throughout the network and are stored in every lymphoid host. The lymphoid hosts are responsible for creating the mobile agents that detect intrusions.

If an intruder successfully attacks a lymphoid host (or otherwise makes the host unreachable), the mobile agents created by that host will still be able to function and carry out their task of intrusion detection. This is possible because once an agent is created, it does not need to communicate with its home platform. Furthermore, when a lymphoid host is disabled, the backup lymphoid host takes over. Therefore, the mobile agents in the network can report intrusions to a lymphoid host even if their home platform has been disabled. Even if all lymphoid hosts are successfully attacked, the mobile agents that remain in the network survive and can still detect intrusions and multiply, while intrusions can be reported to some other host at which the system administrator can be alerted.

For an intruder to disable the operation of the whole IDS, he would need to successfully disable all the lymphoid hosts and all the mobile agents in the network. The lymphoid hosts are static and therefore can be found and disabled by a determined and knowledgeable attacker. However, the mobile agents already in the network are more difficult to disable. To disable the mobile agents, the intruder would first need to find them, but since they are not static and their number can vary, this task is more difficult than for static components (such as the lymphoid hosts). The intruder could also try to attack the agent platforms themselves in an attempt to destroy the mobile agents. However, as every host that we wish to monitor will have an agent platform installed, this task is not trivial if there are many hosts. Therefore, the intrusion detection function is much more difficult to completely disable than in the hierarchical architecture.

The capability of mobile agents to replicate themselves also increases the fault tolerance of the system. An agent can replicate itself without the help of its home platform. Therefore, even if all the lymphoid hosts have been disabled, the number of intrusion detection agents can be increased by agent replication and intrusions can still be detected.

Therefore, the proposed IDS provides greater robustness and fault-tolerance than the hierarchical architectures which nearly all present-day commercial IDSs use. Greater robustness and fault-tolerance are provided because the system can continue to operate and detect intrusions even when most of its components have been attacked and disabled. This capability is not provided in the hierarchical architecture presented in Section 2.2.

6 IMMUNE SYSTEM CONCEPTS APPLIED IN THE IDS

In section 4.2, we have presented properties of the human immune system that make the system fault-tolerant and enable it to effectively detect intrusions. By applying concepts of the immune system to an IDS based on mobile agents, an IDS resulted that also possesses those same properties. The same properties that offered fault-tolerance to the immune system have also contributed to the fault-tolerance of the IDS.

1. *Distributability.* As in the immune system, there is no central controller in the IDS. Intrusion detection is distributed because this function resides within mobile agents that travel from host to host in the network.

2. *Diversity.* A separate database of normal behaviour for each process of interest is created. The database is specific to a particular architecture, software version and configuration, local administrative policies as well as usage patterns. Since there is a large variability in how

individual systems are currently configured and used, the individual databases will provide a unique definition of self for most systems (Forrest et al., 1996).

3. *Disposability*. Any mobile agent of the IDS can be replaced by creating a new one. Therefore, no single mobile agent is essential to detect intrusions.
4. *Autonomy*. Once a mobile agent has been created, it is not directly controlled by some other entity. For example, the agent itself decides when to migrate from one host to another, without being directly controlled by its home platform.
5. *Adaptability*. The system can adapt to detect new types of intrusions because any new behaviour of a program (i.e. behaviour that has not been recorded in the self-database of that program) will be regarded as intrusive. At the same time, previously seen intrusions will also be recognized because they too have not been recorded as part of the normal behaviour of that particular program.
6. *Anomaly detection*. The IDS uses anomaly detection because it is based on identifying deviations in a program's normal behaviour rather than on searching for known attack methods.
7. *Dynamically changing coverage*. In order not to overburden the computer system, not all hosts with an installed agent platform are monitored at the same time. Only those hosts that have a mobile agent running on an agent platform are monitored. Even when a mobile agent is monitoring a particular host, only the program type for which the mobile agent is specialized is being monitored. In addition, the types of programs that are monitored can be changed by changing the types of agents that are present in the network. Therefore, the coverage of programs that are monitored changes dynamically.
8. *Identity via behaviour*. The IDS identifies processes via their behaviour. "Self" processes are identified by the system call sequences that are stored in the self-database of that program. Intrusive or "non-self" processes are identified by the lack of the generated system call sequences in the self-database of that program.
9. *Imperfect detection*. The IDS creates an intrusion alert whenever the behaviour of a process significantly deviates from that recorded in the self-database of that program. Therefore, detection is imperfect because we do not know whether the behaviour deviation is a sign of an actual intrusion or a false alarm.

In conclusion, we can see that the use of mobile agents in the proposed IDS is analogous to the use of white blood cells in the immune system. Mobile agents and white blood cells have several similarities (largely based on Foukia, Hulaas and Harms (2001)). Firstly, both mobile agents and white blood cells are autonomous. That is, once they have been created, they are not directly controlled by other entities (i.e. other organs in the case of white blood cells or other computers in the case of mobile agents). Secondly, both continuously circulate through their domain: white blood cells continually move through the body in the blood, while mobile agents continually move from computer to computer through the network. Thirdly, both are specialized in detecting a particular intrusion. Each white blood cell is specialized to detect only a particular type of antigen (e.g. a specific kind of bacteria) and will not react to another kind of antigen. Similarly, the proposed mobile agents are specialized to detect anomalies in the behaviour of only a particular type of privileged process.

7 CONCLUSIONS AND FUTURE WORK

The purpose of this paper was to present a brief description of how mobile agents and concepts of the human immune system could be applied to increase the fault-tolerance of an IDS. This paper also discussed the main theoretical aspects of how such an IDS can function as well as the similarities between the human immune system and an IDS that uses mobile agents. Certain aspects of the human immune system have not been applied, either because they would complicate the solution or because they were not considered appropriate for an IDS.

Although the proposed IDS provides greater fault-tolerance, it also has some weaknesses. One weakness results from the use of mobile agents. Mobile code has several security concerns that hinder the widespread use of this technology. There are four broad categories of security threats (Jansen & Karygiannis, 1999): (1) agent-to-agent, in which an agent exploits the vulnerabilities of other agents residing on the same agent platform; (2) agent-to-platform, in which an agent exploits the vulnerabilities of its platform; (3) platform-to-agent, in which the agent platform compromises the agent's security; and (4) other-to-platform, in which external entities threaten the security of the agent platform. Ways in which the different threats could be resolved or reduced will need to be investigated. Another weakness results from the fact that the IDS monitors privileged processes. The system will likely miss intrusions where an intruder uses another user's account, as this intrusion class is not likely to be detectable in root processes (Forrest et al., 1996).

Current work involves creating a detailed design of the proposed IDS and describing how it can be implemented. Although this study is still at a theoretical stage, we may in future build a prototype to demonstrate aspects of the IDS to determine its effectiveness and practicality.

8 REFERENCES

- Bace, R., Mell, P. 2001. Intrusion Detection Systems. Special Publication 800-31, National Institute of Standards and Technology (NIST).
- Botha, M. 2004. NeGPAIM: A model for the proactive detection of information security intrusions, utilizing fuzzy logic and neural network techniques. Thesis, Port Elizabeth Technikon, South Africa.
- Bradshaw, J. 1997. An introduction to software agents. In: Bradshaw, J. (Editor), *Software Agents.* AAAI Press/The MIT Press, Chapter 1.
- Forrest, S., Hofmeyr, S., Somayaji, A., 1997. Computer Immunology, *Communications of the ACM*, 40(10): 88-96.
- Forrest, S., Hofmeyr, S., Somayaji, A., Longstaff, T. 1996. A Sense of Self for Unix Processes, In *Proceedings of 1996 IEEE Symposium on Computer Security and Privacy*, pp.120-128.
- Foukia, N., Hulaas J., Harms J. 2001. Intrusion Detection with Mobile Agents. In *Proceedings of the 11th Annual Internet Society Conference (INET 2001)*, Stockholm, Sweden.
- Jansen, W. 2002. Intrusion detection with mobile agents. *Computer Communications* 25 (15): 1392-1401.
- Jansen W., Karygiannis, T. 1999. Mobile agents and security. Special Publication 800-19, National Institute of Standards and Technology (NIST).
- Jansen, W., Mell, P., Karygiannis, T., Marks, D. 1999. Applying mobile agents to intrusion detection and response. Interim Report 6416, National Institute of Standards and Technology (NIST).
- Jansen, W., Mell, P., Karygiannis, T., Marks, D. 2000. Mobile agents in intrusion detection and response. In *Proceedings of the 12th Annual Canadian Information Technology Security Symposium*, Ottawa, Canada.
- Kim, J., Bentley, P. 1999. The Human Immune System and Network Intrusion Detection. In *Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT '99)*, Aachen, Germany.

- Ko, C., Fink, G., Levitt, K. 1994. Automated detection of vulnerabilities in privileged programs by execution monitoring. In *Proceedings of the 10th Annual Computer Security Applications Conference*, pp. 134-144.
- Mukherjee, B., Heberlein, T., Levitt, K. 1994. Network intrusion detection. *IEEE Network*, 8(3): 26-41.
- Rothermel, K., Schwehm M. 1998. *Mobile Agents* In Kent. A. & Williams J. (Editors) *Encyclopedia for Computer Science and Technology* M. Dekker Inc., New York.
- Somayaji, A., Hofmeyr, S., Forrest, S. 1997. Principles of a Computer Immune System. In *Proceedings of New Security Paradigms Workshop*. Langdale, pp. 75-82.
- Spafford, E., Zamboni, D. 2000. Intrusion Detection using Autonomous Agents. *Computer Networks,* 34(4): 547-570.
- Sundaram, A. 1996. An Introduction to Intrusion Detection. *Crossroads: The ACM student magazine*, 2(4).
- White, J. 1996. Mobile Agents White Paper, General Magic Inc.