2006

Building Computer Virus Immune Response using a Bio-mimicry Framework: An innovative and theoretical discourse

Nigel Martin
The Australian National University, Nigel.Martin@canberra.edu.au

John Rice
The University of Adelaide, john.rice@adelaide.edu.au

Building Computer Virus Immune Response using a Bio-mimicry Framework: An innovative and theoretical discourse

Nigel Martin
The Australian National University, Australia
Nigel.Martin@canberra.edu.au

John Rice
The University of Adelaide, Australia
john.rice@adelaide.edu.au

Abstract

While a significant level of research has been dedicated to developing computer anti-virus software using analogies with the human immune system, few development frameworks for the creation of the anti-virus software have been exposed to the broader community of interest. This paper discusses the biological theory related to the human body's immune system and how immune systems might be innovatively mimicked in the development of security systems and software. The paper outlines the Bio-mimicry framework that can be used for scoping the development and features of the security systems and software, including the population of the framework segments. Some commercial security products that are undergoing evolutionary development and current research and development activities are used to augment the development framework and explicate its use in practice.

Keywords

Security, architecture, information, innovation, networks, standards.

INTRODUCTION

In 1983 a series of five controlled viral attack experiments conducted by a promising young doctoral student at the University of Southern California proved the concept of a 'computer virus' (Cohen 1985). Since that early period, it has been observed that computer viruses have evolved into pieces of software code that exhibit two specific characteristics (Hoffman 1990, Ludwig 1996). First, the code has a partial or fully automated capability to reproduce or clone itself. Second, the code can transport itself by attachment to a computing entity (such as a program, disk sector, data file) and ensuing transfers between the various system entities. In the years that followed the seminal research and experimentation, the information systems community has attempted to dissect and develop a greater understanding of computer viruses (Cohen 1987, Hoffman 1990, Ferbrache 1992, Cohen 1994, Ludwig 1996, Szor 2005). In essence, computer scientists and software experts have attempted to understand the pathology of computer viruses, or their basis as an artificial life form (Ferbrache 1992, Spafford 1994). Whether the code takes the form of an add-on virus that attaches itself to host programs or software, is an intrusive virus that overwrites the host code, or takes on a polymorphic structure that continues to replicate itself and infect large networks, the quest for greater understanding in this important area of computing security continues.

The parallels drawn with biological hazards, viruses and immune system response has lead to a substantial level of research in the areas of software modelling, biological systems based design, anti-virus architectures, viral software testing and analysis, and computing heuristics. Some researchers have conducted a matched analysis between human and artificial (computing) immune systems, identifying important similarities (and notable differences) between the immune systems, and describing desirable features that should be mirrored into artificial environments. For example, Skormin et al (2001) identified that both systems were highly complex, distributed and connected with many entry points, were vulnerable to intentional or unintentional introduction of foreign bodies, and must be capable of detecting and neutralising alien matter. Similarly, Harmer et al (2002) asserted that both systems must maintain a massively parallel and distributed architecture for communications and signalling, be capable of self/non-self determination, support autonomic behaviours in attacking new foreign matter and infections, and invoke memory based responses to attacks from past infections. Other research has concentrated on using the biological immune system as an inspirational model for computer anti-virus software (Kephart 1994, Kephart 1995, Forrest et al 1997, King et al 1999, Goel and Bush 2004; Goldenberg et al 2005). The concepts of innate and adaptive biological immune systems are used as direct physical models for developing virus pattern recognition, computer immunological memory, and autonomic virus patch software. Given the evolving business environment where malicious software threats (ie, worms, viruses, infectious agents) are becoming commonplace, the development of virally immune self healing or self defending information systems networks appears to hold some promise.

In exploring this line of inquiry, a review of biological immunity literature suggests that the development of secure networks and software that mimics the human immune system may yield substantial benefits for the protection of critical information and communications technology infrastructure. However, an immune system response to computer viruses and worms would likely involve screening for abnormalities, quarantining the infectious agents, and developing software antibodies to combat the destructive agents. This raises the question: What type of development framework can software organisations use to create security systems and anti-virus software? This paper presents an innovative development framework that uses biological models for the analysis and creation of artificial systems (Benyus 1997).

A detailed explanation and summary of the human immune system, including the types of immunity and the biological delivery mechanisms, serves as a theoretical platform for the system development discussion. It is considered important that a comparison and contrast of the biological and information systems immunity problem space be conducted, including the treatment of viruses and virus mutations in both domains. We then emphasize that the development of security systems and software using a biological lens may prove more successful than the current practices and processes. By adopting the biological viewpoint, and describing the bio-mimicry terminology and theory, a discussion of some specific examples of how the mimicking of biological systems has supported the solving of human problems (eg, deep sea sponge structures used as biological models for fibre optic strand development by Lucent Technologies) is developed. The paper then explicates the bio-mimicry framework and populates the framework with the structure for developing security systems and software, including computer virus immune response. The framework is augmented using examples from current research efforts and developments in the area of information systems network immunity and some commercially available network protection software systems. The paper concludes with some further Information and Communication Technology (ICT) development opportunities that might be pursued using the bio-mimicry framework.

INFORMATION SYSTEMS NETWORK IMMUNITY – A THEORETICAL PLATFORM

The following sections discuss the technical concepts of human and biological immunity, the spread of viruses and infectious agents in the biological and information systems domains, and the potential for the successful development of security systems and software using biological models.

Human Immune Systems

The human immune system is a complex network of specialised cells and organs that protects the body from external biological influences and conditions. Importantly, the immune system provides this protection by responding to antigens (normally large molecular proteins) that gather on the surface of infected cells, viruses, bacterial agents or other pathogens. A large genomic region in our bodies known as the Major Histocompatibility Complex (MHC) contains special genes with critical immune system functions (ie, the Human Leukocyte Antigen (HLA) genes). These HLA genes encode cell surface antigen presenting proteins, as part of the normal cellular structure. This encoding process allows the immune system to use HLA to differentiate between "self" and "non-self" cells. Any cell displaying that individual's HLA type is identified as 'self' (no immune response) with cells displaying another HLA type identified as 'non-self ' (immune response) (Roitt et al 2001, Paul 2003, Doherty 2003).

The human immune system is bifurcated into two major components, Innate immunity and Adaptive (or acquired) immunity. Innate immunity includes the barriers that isolate harmful or foreign bodies as a first line of immune defence (eg, skin, mucus, stomach acid). The innate system also includes white blood cells, commonly known as phagocytes, that destroy micro-organisms and dead and damaged cells. Innate system phagocytes work by surrounding, engulfing and finally destroying the foreign substances or pathogens. In contrast, the adaptive immune system is based on white blood cells (termed leukocytes) that are produced by stem cells in the bone marrow, and ultimately mature in the thymus gland and/or lymph nodes of the body (Roitt et al 2001, Paul 2003, Doherty 2003).

The adaptive immune system can be partitioned into two further protective sub-systems (Roitt et al 2001, Paul 2003, Doherty 2003). The first sub-system is the *Humoral* immune system. Under this immune system, a special type of leukocyte termed *B Lymphocytes* (or B cells) are formed in bone marrow and produce antibodies (termed *immunoglobulins*) that bind to the specific bacteria or virus, thereby making it easier for the phagocytes to target and kill the antigens. The second sub-system is the Cellular immune system that destroys virus infected cells with *T Lymphocytes* (also known as thymus cells or T cells). *Cytotoxic or Killer* T cells ($CD8^+$ T cells) identify infected cells by using receptors to scan the cell surface. $CD8^+$ T cells release granzymes that trigger apoptotic ('suicidal') behaviour, thereby killing that cell and any viruses it may be creating. *Helper* T cells ($CD4^+$ T cells) activate a specific form of phagocyte termed *Macrophages* that ingests the dangerous material, while also producing proteins known as cytokines (interleukins) that induce the proliferation of B and development of T cells (Doherty 2003).

Biological and computer viruses

Biological viruses are microscopic parasites that infect the cells of biological species and organisms. Viruses are obligate intracellular parasites that reproduce and replicate by invading and controlling other cells. Importantly, these types of parasites do not have self-reproduction machinery and tend to infect single and multi-celled organisms. Viruses typically carry a small amount of nucleic acid surrounded by a protective coating of proteins, lipids, glycoproteins or a combination of these substances known as a capsid (Roitt et al 2001, Paul 2003).

Comparatively, a computer virus is an executable program that can replicate itself by invading a host (much like a biological virus), and spreading to other devices as the host is shared or exchanged amongst the device population (Ferbrache 1992, Spafford 1994). The growing portability of computing and wireless communication devices is providing expanding opportunities for the transfer of viruses and infected agents. Additionally, viruses may spread through multiple devices accessing network file systems. The most common type of virus is the file virus that infects files or program libraries on an operating system. Macro viruses can be hidden in embedded macros within documents and can self execute when the file is opened, while boot viruses infect the boot sector of diskettes or the master boot record of a hard disk.

Computer worms and Trojan horses are other forms of malicious software that have evolved from the early viruses (Szor 2005). A computer worm is a self-replicating form of program that is similar to a virus. However, a worm is self-contained code and does not need to be part of another program to propagate across the network. Worms are configured to utilise the file transmission capabilities of computers and network devices, and issue copies of the worm program to other system components. Also, worms often consume large segments of network bandwidth and materially damage the performance of the network and business environment.

Trojan horses take the form of legitimate software programs and perform undesirable technical functions. The functions generally have a malicious intent including spying and backdoor access, which may allow the computer to be remotely controlled (also known as a "zombie" terminal). Advances in the construction of Trojan horse programs have allowed these types of software to replicate through the invasion of a host program or system. This type of evolution has meant that current Trojan horses act much more like viruses, and are generally more infectious than in the previous forms.

Using biology to develop security systems and software

Experts in the field of computer viruses and malicious software have noted that:

"Natural immune systems protect animals from dangerous foreign pathogens, including bacteria, viruses, parasites, and toxins. Their role in the body is analogous to that of computer security systems in computing. Although there are many differences between living organisms and computers, the similarities are compelling and could point the way to improved computer security." (Cohen 1987, Forrest et al 1997)

This analogy suggests that biological and computer viruses share many of the same technical characteristics (eg, spread through host agents and systems, take mutated forms, highly infectious) and conventions (eg, strain identification and nomenclature). A good example of common biological and computer virus convention is viral identification schemas. The identification of the various hepatitis viruses by strain and alphanumeric nomenclature shares similar features with the identification tags placed on malicious "Nimda" and "Sasser" computer worms as shown in Table 1.

Biological Virus ID	Computer Worm 1 ID	Computer Worm 2 ID
Hepatitis A Virus	W32.Nimda.A@mm ; W32.Nimda.A@mm(dll)	W32.Sasser.B.Worm
Hepatitis B Virus	W32.Nimda.A@mm(dr) ; W32.Nimda.A@mm(html)	**W32.Sasser.C.Worm**
Hepatitis C Virus	**W32.Nimda.B@mm(dll)** ; **W32.Nimda.B@mm(dr)**	W32.Sasser.D
Hepatitis D Virus	W32.Nimda.C@mm	**W32.Sasser.E.Worm**
Hepatitis E Virus	**W32.Nimda.corrupt**	W32.Sasser.F.Worm
	W32.Nimda.E@mm ; W32.Nimda.E@mm(dr)	**W32.Sasser.G**
	W32.Nimda.enc ; **W32.Nimda.enc(1)** ; **W32.Nimda.enc(dr)**	W32.Sasser.gen.Worm
	W32.Nimda.I@mm	
	W32.Nimda.J@mm	
	W32.Nimda.K@mm	
	W32.Nimda.M@mm	
	W32.Nimda.N@mm	
	W32.Nimda.P@mm	
	W32.Nimda.Q@mm	
	W32.Nimda.R	

Table 1: Summary of Hepatitis biological virus and Nimda and Sasser computer worm identifications (Symantec AntiVirus 9.0.3.1000, 15 January 2006, Revision 8)

Given the similarities between the biological and computer viruses, the development of security software and systems and computer immune responses might follow parallel pathways. For example, antivirus systems might be designed to act like innate phagocytes where the malicious code, on entry into the environment, is "surrounded and neutralised". In a similar manner, a new design might include a B Lymphocyte type behaviour where remedial code is "attached to the computer virus" making the virus easier to identify and neutralise.

These types of design concepts suggest that the issue of virus outbreak lead times would present fewer problems for security analysts. Rather than designing an antivirus patch (following identification of a vulnerability and publication of the exploit code by programmers and hackers) in anticipation of a viral outbreak, a self-healing or immune network would allow the infection to be identified and neutralised upon entry (Bekker 2003). The outbreak of computer viruses during 2001-2004, and the short patch deployment lead times as shown in Table 2, demonstrates that a self-defending network security paradigm may have been more effective than current design practices (Cisco 2005).

Computer Virus ID	Patch ID	Patch Availability Date	Virus Outbreak Date	Total Lead Time
Nimda worm	MS00–078	17 October 2000	18 September 2001	336 days
Slammer worm	MS02–039	24 July 2002	25 January 2003	185 days
MSBlaster.A worm	MS03–026	16 July 2003	11 August 2003	26 days
Sasser.A worm	MS04–011	13 April 2004	30 April 2004	17 days

Table 2: Examples of computer virus and worm outbreaks 2001-2004 (Merkl 2004)

BIO-MIMICRY TERMINOLOGY AND THEORY

The following sections discuss the theory and various terms that relate to bio-mimicry and the associated development framework. This section also provides some tangible examples of biological models that have been used to solve complex human problems, and develops a populated framework for the development of security software and systems for the treatment of viruses and network infections.

Bio-mimicry – Using biological models to solve complex human problems

Bio-mimicry is the scientific discipline that studies the best concepts in nature and biology and imitates these types of designs and processes in order to solve complex human problems (Benyus 1997). The Bio-mimicry term has latin roots with 'bios meaning life', and 'mimesis meaning to imitate'. The discipline is based on the premise that nature and biological species have efficiently solved several problems that humans are still looking to resolve. Some examples of bio-mimicry being used to solve complex human problems are outlined as follows:

- The Defense Advanced Research Projects Agency (DARPA) of the United States Department of Defense, and the National Aeronautics and Space Administration (NASA) are conducting a joint study of the navigational systems and locomotive strategies of insects and entomological species in order to design the next generation of autonomous robots and space exploration vehicles.
- University of Leeds researchers are studying the jet-based defence mechanism of the bombardier beetle to determine whether the insect can assist them in designing a re-ignition system for a gas-turbine aircraft engine in mid-flight. The beetle is capable of spraying potential predators with a high-pressure stream of boiling liquid excreted at 100 degrees Celsius.
- Nanotechnology researchers at the Massachusetts Institute of Technology (MIT) are attempting to understand the soft-bodied structures of sea snails and other like creatures in order to develop light weight armor systems for soldiers, police and other law enforcement officers. The MIT scientists are studying the structure and mechanics of the tough inner layer of mollusk shells called 'nacre' or mother-of- pearl at extremely small nanometer-length scales (one billionth of a metre).

The Bio-mimicry development framework

The Bio-mimicry development framework is composed from a series of actions and questions that guide the design of new systems, devices and mechanisms (Bio-mimicry Guild 2005b) and is depicted in Figure 1.

1	**Identify the problem** (What do you want the design to do?) **Ask the important "Why?" questions** (eg, Why do the current systems fail?)	
2	**Place the Question in a Biological Frame** Identify all the functions. Define the operating parameters and conditions. How are those functions delivered/not delivered in natural systems?	**Define environmental (operating) parameters and conditions** Identify the climatic conditions. Define the nutrient (power source) requirements. Identify the social parameters and interactions. Record the temporal conditions and events.

3	**Find the best biological or natural models** (Consider the literal and metaphorical models. Undertake a literature search in the area of interest. Consult experts in the allied biological field of interest.)	
4	**Create a taxonomy of design strategies** (Prioritise the most promising strategies for emulation given the operating conditions and design parameters.)	
5	**Develop a "sandbox play" area and develop designs** (Is the design modelling form, process or system? Understand the scale and scope effects. Consider the influencing factors on the effectiveness of the processes and systems.)	
6	**Review the design against the biological model principles** Does the design create conditions for continuous lifecycle operations?	Is the design modular/segmented? Is the design built to shape? Is the design self-assembling? Is the design cyclic? Can the design detect feedback, adapt and/or evolve? Is the design useable? Will users find it easy to use?

Figure 1: Bio-mimicry development framework (Bio-mimicry Guild 2005b)

The first part of the framework asks the designer to identify the problem space and outline the important 'why' questions (eg, Why do the current systems fail? Why do some computer viruses appear impervious to firewalls?). The second part of the framework requests that the designers place the problem in a biological frame (or lens) and define the operating parameters and conditions, including the prevailing climate, social interactions, and temporal conditions and events. The third part of the framework asks that designers examine and select the best biological and natural models for their functional designs. This may include detailed discussions with experts in the allied biological field of interest (eg, immunology, virology, parasitology). The fourth part of the framework allows the designers to make value judgements and trade-off decisions in developing a prioritised taxonomy of designs. The fifth part of the framework facilitates further development of the designs through testing and sandboxing, including understanding the effects of scale/scope and influential design factors. The final part of the framework is a design review that compares the solution with the biological model's shape, characteristics and functions.

Using biological immune system models to develop security software and systems – A populated framework

The following sections provide summaries of the bio-mimicry framework segments (parts 1-6) as applied to the development of security software and systems using biological immune system models. The development steps are augmented with examples from the current base of literature and commercial system development activity.

Part 1 - Defining the problem

The problem is best defined as:

"The development of a self healing (or defending) network that is capable of an active immune response to any introduced computer virus, worm, or other evolving forms of infection".

The reasons behind developing these forms of virus immune networks include the increase in network security threats (through hacking and intrusion), the present inefficient system development paradigm that depends on building antivirus scripts in anticipation of a security event or incident (noting the decreasing lead times), irregular updates of antivirus software by users and clients, high rates of re-infection from un-patched terminals and devices over extended periods of time, and the limited availability of dedicated vendor and user resources for real-time security patch development and proactive deployment (Somayaji et al 1997, Chen and Robert 2004, Dasgupta 2004).

Part 2 – Identify functions and define environmental parameters and conditions

The functions of the security systems and software should include the capability to "detect" abnormalities in the network's operations and systems, "isolate" the computer virus and/or infections, and "develop" software antibodies that "neutralise" the viral effects through "destruction" of the malicious code or rendering the code ineffectual through mediation induced behaviours (Kephart and Arnold 1994, Kephart et al 1997, Chen and Robert 2004).

These types of functions are delivered in biological settings in the form of human and animal immune systems and include the functions for engulfing and destroying infected cells and foreign substances, the generation of antibodies that facilitate and assist virus eradication, and cellular mediation that modifies the infected cell's behaviours (eg, cellular self destruction or apoptosis) (Roitt 2001, Paul 2003).

The operating environment in which computer viruses and infections can be encountered includes dynamic local and wide area computing and communications networks, with complex arrays of operating systems, software

applications, and databases, coupled with a broad range of system hardware and devices (Bradley and Tyrrell 2001). These types of computing environments tend to have temperature and air quality controls with multiple users in various locations. Administrative procedures and normal daily network operations suggest that users are continuously added and removed from the networks, while users concurrently access various applications and datasets.

Part 3 – Biological or natural models

In this bio-mimicry framework exercise, the human immune system has been selected as the "default" best biological model on which to base the proposed security systems and software solution (Biomimicry Guild 2005a). Other biological or natural system models may provide an equivalent level of utility for this form of system development (eg, the use of anti-venom treatments for neutralisation of poisonous snake and spider bites).

Part 4 – A taxonomy of designs

The taxonomy of designs for this bio-mimicry exercise may include "identify-surround-neutralise" (phagocyte), "identify-attach-neutralise" (antibody) and "self destructive" (apoptotic) virus immunity systems and software. Typical priorities (based on the likelihood of successful product development) that could be applied to the designs might be phagocytic (1), antibody (2), and apoptotic (3), where phagocytic designs may prove the most successful of all the systems developed, while apoptotic designs may provide greater social and technical challenges in the immediate term. These ratings serve only as examples, and would typically be based on expert opinions provided by antivirus software developers and vendors. Figure 2 depicts the design schemas for the proposed systems.

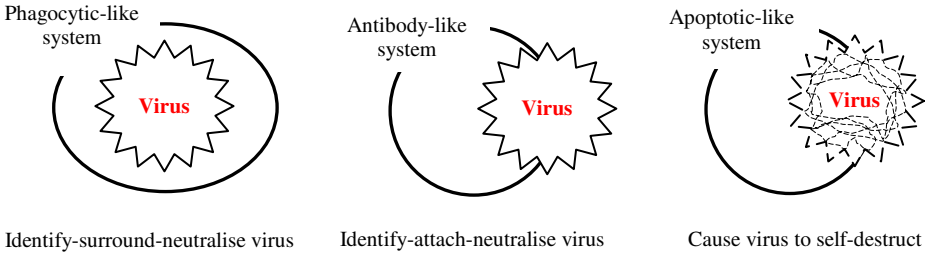

Figure 2: Design taxonomy – schemas for phagocytic, antibody and apoptotic mimicked immunity programs

Part 5 – Sandbox and design development

In this bio-mimicry framework exercise, no sandbox area has been designated for system prototype design and testing. However, in current international research and development activities, computer virus test bed environments are available. Good examples of the test environments are the Internet Technology Laboratory test bed at the University of Arizona (Hariri et al 2003), the sand-boxed test environment at Columbia University (Sidiroglou and Keromytis 2005), and IBM's High Integrity Computing Laboratory (Kephart et al 1997). These test environments would allow the system and software designers to evaluate the detection range of introduced viruses and infections, speed of delivery and dissemination of anti-virus prescriptions, and scalability factors such as reduced data rates and vulnerable system components. Importantly, these laboratory environments would support the critical fifth part of the bio-mimicry based system development.

Part 6 – Design review

In this bio-mimicry framework exercise, no formal design has been developed and accordingly no design review conducted. However, a number of commercial computer immune systems products, such as Microsoft's Network Access Protection (NAP) and Cisco Systems' Network Admission Control (NAC), provide examples for a simulated review (Cisco Systems 2005, Microsoft Corporation 2005). The NAP and NAC products form part of the network quarantine group of technologies. These products monitor, assess and isolate system components (eg, personal computer terminals, servers, and personal digital assistants) that increase network vulnerability through their possession of non-compliant antivirus programs, out-of-date virus signatures, or un-patched applications and operating systems. The products take a "reverse approach" to traditional antivirus technologies (eg, Symantec Antivirus) by quarantining vulnerable or infected systems and components rather than attacking the computer virus itself. In this example, the products possess some detection functions, but clearly do not display the more direct virus and infection isolation, neutralisation or destruction functions established under part 2 of the development framework. Consequently, the part 6 review may usefully identify a number of functional variations or deficiencies when compared with the biological or natural system models.

CURRENT RESEARCH AND DEVELOPMENT – COMPUTER NETWORK IMMUNITY USING BIOMODELS

Some specific high profile activities demonstrate that the commercial and research communities of interest are presently investing in the research and development of security systems and software that mimics biological immune systems. First, the United States Army Research Office has provided the Electrical and Computer Engineering Department at the University of Arizona with a US$1 million grant to develop bio-mimicked security software. The software is scoped to screen information technology networks for abnormalities, isolate infectious viruses and worms, while developing coded antibodies to fight infections. The first part of the research program will establish the rudimentary modelling techniques and tools, while the second part of the research will be focused on implementing the antiviral techniques (Stiles 2005). In the second example, the Electronics Department at the University of York has established a funded artificial immune systems research network, comprising of over 125 computer related academics and professionals, under its Bio-inspired Architectures Laboratory. The network supports researchers in establishing the collaborative infrastructure to drive forward research in the areas of computer system immunity, fault tolerant hardware systems, and active machine learning (Network for Artificial Immune Systems 2005). These activities serve as important examples of the innovative use of biological models for researching and developing computer system immunity.

CONCLUDING STATEMENTS

Innovation drives product research and commercialization down many paths that may not have been necessarily explored given the often conventional approaches adopted by system designers and engineers. The use of biological and natural system models in the development of artificial and man made systems and products could certainly be characterized as technically and managerially innovative. Examples presented earlier in this paper demonstrate the value and utility of the approach in solving complex human problems.

In this paper we have presented the theoretical platform relating to biological immune systems and drawn parallels with computer network immunity and antiviral approaches. Our introduction and explication of the Bio-mimicry framework as a system development tool provides a different and innovative dimension to the development of artificial immune systems. The Bio-mimicry framework comprises six parts or steps that allows system designers and developers to define the problem, analyse and identify the desired functions, select the premium biological model, develop and sandbox test the taxonomy of designs, and review the outturn systems or products. The framework enables a different set of thought processes when compared to the predominantly technical and mathematical literature related to computer network immunity.

This paper also demonstrates the viability of the framework through our augmentation approach. This includes our use of expert opinion in extant literature, identified system functions and desired characteristics, and commercial computer immunity products, in populating parts of the framework. Finally, while some current research programs are exploring the use of bio-mimicry for computer system immunity, other opportunities for developing bio-inspired information technology exist. As an example, the development of “self healing” optical fibre remains one of the biggest unsolved problems within the telecommunications industry. Damage to the fibre due to earthworks and unauthorized site excavation presents a common maintenance problem for telecommunications providers. A bio-mimicked fibre material or technology might be developed to solve this, and other similar ICT problems.

REFERENCES

Bekker, S, (2003). Appearance of Exploit Code means time is running out to apply critical windows patch. Enterprise Magazine. Available at http://www.entmag.com/news/ article.asp?EditorialsID=5953 (accessed at 19 December 2005)

Benyus, J.M., (1997). Biomimicry: Innovation Inspired by Nature. William Morrow and Co., Inc., New York.

Biomimicry Guild (2005a). Biomimicry: An Introduction. Available at http://www.biomimicry.net/biom_project.html (accessed at 2 November 2005).

Biomimicry Guild (2005b). Evolving Biomimicry Methodology. Available at http://www.biomimicry.net/essent_resourc.html (accessed at 17 January 2006).

Bradley, D.W. and Tyrrell, A.M., (2001). The Architecture For A Hardware Immune System, The Third NASA/DoD Workshop on Evolvable Hardware.

Chen, T.M, and Robert, J.M, (2004). Worm Epidemics in High Speed Networks. IEEE Computer, June, 48-53.

Cisco Systems, (2005). Network Admission Control. Available at http://www.cisco.com/en/US/netsol/ns466/networking_solutions_package.html (accessed at 1 October 2005).

Cohen, F.B, (1985). Computer Viruses. University of South California.

Cohen, F.B, (1987). Computer Viruses: Theory and Practice, Computers & Security, 6, February, 22–35.

Cohen, F.B., (1994). A Short Course on Computer Viruses, 2nd ed. Wiley, New York.

Dasgupta, D, (2004). Immuno-Inspired Autonomic System for Cyber Defense. Computer Science Technical Report, May.

Doherty, P., (2003). Sir John Eccles Centenary Lecture, University of Mebourne, 18 March 2003.

Ferbrache, D., (1992). A Pathology of Computer Viruses. Springer-Verlag, Berlin.

Forrest, S, Hofmeyer, S.A., and Somayaji, A, (1997). Computer immunology, Communications of the ACM, 40, 10, October, 88–96.

Goel, S. and Bush, S.F., (2004). Biological Models of Security for Virus propagation in computer networks. Login, 29, 6, December, 49-56.

Goldenberg, J, Shavitt, Y, Shir, E, and Solomon, S, (2005). Distributive immunization of networks against viruses using the 'honey-pot' architecture. Nature doi:10.1038/nphys177, December.

Hariri, S, Guangzhi Qu, Dharmagadda, T, Ramkishore, M, and Raghavendra, C.S., (2003). Impact Analysis of Faults and Attacks in Large-Scale Networks. IEEE Security and Privacy. September-October, 49-54.

Harmer, P.K, Williams, P.D, Gunsch, G.H, and Lamont, G.B, (2002). An Artificial Immune System Architecture for Computer Security Applications. IEEE Transactions on Evolutionary Computation, 6, 3, June, 252-280.

Hoffman, L. J., (1990). Rogue Programs: Viruses, Worms, and Trojan Horses. Van Nostrand Reinhold, New York.

Kephart, J.O., (1994). A biologically inspired immune system for computers, Proceedings of the. Fourth International. Workshop Synthesis and Simulation of Living Systems, July 1994, 30–39.

Kephart, J.O., (1995). Biologically Inspired Defenses Against Computer Viruses, Proceedings of IJCA '95, 985–996.

Kephart, J.O., and Arnold, W.C., (1994). Automatic extraction of computer virus signatures, Proceedings of the 4th Virus Bulletin International Conference, R. Ford, Ed. Virus Bulletin Ltd., Abingdon United Kingdom, 179–194.

Kephart, J.O., Sorkin, G. B., Swimmer, M., and White, S. R., (1997). Blueprint for a computer immune system, Proceedings of the Virus Bulletin International Conference. Virus Bulletin Ltd., Abingdon United Kingdom.

King, R.L., Lambert, A.B., Russ, S.H., and Reese, D.S., (1999). The Biological Basis of the Immune System as a Model for Intelligent Agents. Proceedings of the 11 IPPS/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing, 156-164.

Ludwig, M. A., (1996). The Little Black Book of Computer Viruses. American Eagle, Show Low, Arizona.

Merkl, W., (2004). Self Defending Networks. Cisco Systems. Available at http://www.cisco.at/partner/pdf/wmerkl-7423.pdf (accessed at 15 June 2005).

Microsoft Corporation, (2005). Network Access Protection. Available at http://www.microsoft.com/technet/itsolutions/network/nap/default.mspx (accessed at 1 December 2005)

Network for Artificial Immune Systems, (2005). University of York. Available at http://www.artificial-immune-systems.org/artist.htm (accessed at 7 February 2006).

Paul, W.E., (2003). The Immune System: An Introduction, Fundamental Immunology, 5th Ed., Raven Press Ltd, New York.

Roitt I., Brostoff J., and Male D. (2001). Immunology. 6th edition. Gower Medical Publishing, London.

Sidiroglou, S, and Keromytis, AD, (2005). Countering network worms through automatic patch generation. IEEE Security and Privacy. November-December, 52-60.

Skormin, V.A., Delgado-Frias, J.G., McGee, D.L., Giordano, J.V., Popyack, L.J., Gorodetski, V.I., and Tarakanov, A.O., (2001). "BASIS: A Biological Approach to System Information Security," Proceedings from Mathematical Methods, Models, and Architectures for Network Security Systems (MMM-ACNS) Conference 2001, 127–142.

Somayaji, A, Hofmeyer, S, and Forrest, S, (1997). Principles of a computer immune system, Proceedings of New Security Paradigms Conference, September, 75–82.

Spafford, E.H., (1994). Computer Viruses as Artificial Life. Journal of Artificial Life, MIT Press, Boston.

Stiles, E, (2005). UA ECE Gets $1 Million to Fight Cyberspies With Bio-Mimicking Software. University of Arizona News Release (28 October 2005). Available at http://news.mongabay.com/2005/1206-ua.html (accessed at 15 December 2005).

Szor, P., (2005). The Art of Computer Virus Research and Defense, Addison-Wesley Professional, Boston.

ACKNOWLEDGEMENTS

The authors wish to thank the John Curtin School of Medical Research at The Australian National University for their assistance in undertaking this research and preparing this paper.

COPYRIGHT