Journal of Computer Science 2 (10): 785-788, 2006
ISSN 1549-3636

Analysis of Virus Algorithms

[1]Jyoti Kalyani, [2]Karanjit Singh Kahlon, [3]Harpal Singh and. [4]Anu Kalyani
[1]CBM Department, GNDU Amritsar, Punjab, India
[2]Department of Computer Science and Engineering, GNDU, Amritsar, Punjab, India
[3]Department of Computer Science, LIM, Jalandhar, Punjab, India
[4]Departement of Computer Science, Punjab Technical University, Jalandhar, Punjab, India

Abstract: Security of wired and wireless networks is the most challengeable in today's computer world. The aim of this study was to give brief introduction about viruses and worms, their creators and characteristics of algorithms used by viruses. Here wired and wireless network viruses are elaborated. Also viruses are compared with human immune system. On the basis of this comparison *four guidelines* are given to detect viruses so that more secure systems are made. While concluding this study it is found that the security is most challengeable, thus it is required to make more secure models which automatically detect viruses and prevent the system from its affect.

Key words: TSR, viruses, worms, malware

INTRODUCTION

Virus programs use the most basic computer functions and operations like copying, deleting and automatic operations like decision making. It should also be clear that no additional functions are necessary for the operation of viral programs. Because of these characteristics it is difficult to differentiate between virus program and valid program. Without running the program, or simulating its operation, there is no way to say that this program is viral and that one is valid. It will focus on understanding how virus writers operate, how they perceive their world and the world around them and how they think. Broadly virus writers can be categorized according to *three stages*:

Stage I: In early times, viruses were written by young programmers who had just learned programming. They try to use their skills[1]. But viruses created by such writers do not spread because of disk reformation or system up gradation. Viruses written by these writers are not written for some specific purpose but only to show their talent. They were still at their learning stage, but had already made a conscious decision to devote their skills to virus writing. They were people who had chosen to disrupt the computing community by committing acts of cyber hooliganism and cyber vandalism[2]. Viruses written by members of this group were usually extremely primitive and the code contained a large number of errors. They learn new techniques and share their views with professional virus writers through chat rooms or emails.

Stage II: And then these young programmers grew up. Not all of them grew up, but rest of them becomes professional virus writers who have the ability to create code to harm several computers and networks. After some time this group becomes most dangerous section of the computer underground[3]. After creating destructible softwares and hardwares, their main aim is to spread their creations and to ensure whether their creations are spreading, they use social engineering.

Stage III: Actually virus writers are programmers who feel themselves as researchers but this research is illegal. They use their ability to harm the community. Their motive behind writing these infected programs is fair because they do it for their research purpose only not for money[4]. They do not spread their creations like viruses but they do discuss innovations on the internet. Still there are researchers who are actually working to detect and remove viruses. They create different patches and anti viruses to detect and prevent viruses.

FEATURES OF VARIOUS VIRUS ALGORITHMS

Remember virus is a program. Therefore each virus has different code and algorithm. These virus algorithms can be differentiated according to their features as each algorithm is designed for some specific task.
Features of operating algorithms:

* Ability of virus to cover traces
* Use of Self encryption
* Polymorphic capability
* Metamorphic code Algorithm
* Terminate and Stay Resident capability
* Use of non-standard techniques

Ability of virus to cover its traces with the help of Stealth algorithm viruses can completely or partially cover their traces inside the Operating System. A Stealth virus gets its name because it is very difficult to find and it uses various complex techniques for

Corresponding Author: Jyoti Kalyani, CBM Department, GNDU Amritsar, Punjab, India, E-mail: jyoti_kal@yahoo.co.in

avoiding detection. The virus has the ability to redirect system pointers and information in order to infect a file without actually changing the infected program file. Another Stealth technique is to conceal an increase in file size by displaying the original uninfected file size[5]. The most common Stealth algorithm is interception of OS read/write calls to infected objects. In such cases Stealth viruses either temporarily cure them or substitute themselves with uninfected pieces of information. In case of macro viruses the most popular technique is to disable the View Macro menu(s). Frodo is one of the first file Stealth virus, Brain is the first boot Stealth virus[6].

Use of self encryption algorithm This is more advanced method in which virus uses simple encryption algorithm to encipher itself. Here, the virus consists of a small decrypting module and an encrypted copy of virus code. For each infected file virus is encrypted with different key but decrypting module remain the same. Therefore, a virus scanner cannot directly detect the virus using signatures but it can still detect the decrypting module which still makes indirect detection of virus possible[7]. Mostly, the decrypting techniques that these viruses use are very simple and can be done by just xoring each byte with randomized key that was saved by parent virus. The advantage of using xor-operations is that the encryption and decryption routines are same.

Polymorphic code was the virus algorithm that posed a serious threat to virus scanners[8]. Just like self encrypted viruses, a polymorphic virus infects files with an encrypted copy of itself, which is decoded by a decryption module. But in the case of polymorphic viruses however, this decryption module is also modified on each infection[9]. Therefore, polymorphic virus has no parts that stay the same on each infection, making it impossible to detect directly using signature.

Metamorphic code Algorithm This virus has distinguished characteristic i.e. every time it changes its code completely to infect any executable file. Viruses that use this algorithm are called metamorphic viruses. It require metamorphic engine to enable metamorphism. These virus programs used to be very large and complex e.g. W32/Smile is metamorphic virus consist of 14,000 lines out which 905 code is part of metamorphic engine.

TSR stands for *terminate but stay resident* This virus will remain resident in your computer's memory after it executes. There are number of viruses, particularly boot sector viruses, which remain resident in memory so that they can spread to other disks and programs much faster and more transparently[10]. It is very difficult to find the virus if it has become the memory resident because it can monitor every action taken by your computer and cover its traces accordingly. When TSR virus infect any system it leaves its resident part in RAM which then intercepts system calls to target objects and incorporates into them. These resident viruses stay in memory and are used to be operative until power down or until operating system reboot. But nonresident viruses not at all infect computer memory and are active for a limited time only. Some nonresident viruses leave their small resident parts in RAM which do not spread the virus still such viruses are called nonresident[11]. Some other viruses like macro viruses can also be considered residents because they reside in computer memory during all the run time of the infected editor program. Here the editor plays the role of operating system and system reboot means the editor program termination. In multitasking operating systems, the lifetime of a resident DOS virus can also be limited by the moment of closing of the infected DOS window, the activity of boot viruses in some operating systems is limited to the moment of installation of OS disk drivers.

Nonstandard techniques Viruses uses many nonstandard techniques to avoid detection in OS kernel to protect its residents copy from being detected and make curing more difficult.

PROPOSED GUIDELINES FOR CONTROLLING VIRUSES ON THE BASIS OF HUMAN IMMUNE SYSTEM

Human immune system is itself a network which consists of human webs, sexual webs, food webs etc. So these are the transmission mediums for spreading viruses but in computer system technological networks exist such as internet, email which transports computer viruses. Humans are self-regulating against viruses, while computers are not. That is why strong immune system is required for virus control. Biological viruses and computer viruses both have different level of sophistication like biological viruses are autonomous, evolving and sequential whereas computer viruses are highly regulated and static. In general, biological viruses are less connected than computer viruses. There are various analogies between biological and computer viruses. Based on principles extracted from mapping between computer system and immune system, some guidelines are proposed for computer security which is given as follows:

Data protection: Computer viruses are the programs which infect programs or boot sectors by inserting instructions into program files stored on disk. According to this definition of viruses, the protection problem is essentially the same as that of protecting any kind of stored data. Many change-detection algorithms have been devised to address this problem including some inspired by biology. Immunization exists here also; they are patches, alerts, virus scanning and OS updates etc. Here an antibody counter measure corresponds to virus scanner, which acts like antibody cells for the protection of data.

Single host protection (active processes): Suppose every active process in computer is cell as similar as adaptive human immune system which is made up of cells which monitor and interact with other cells. Then it can be said that a computer runs multiple processes as a multicellular organism and network of such computers can be considered as a population of such organisms. Different security mechanisms, such as passwords, groups and file permissions etc. would protect the computer analogous to that of a computer's skin and innate immune system[12]. With the help of lymphocyte an adaptive immune system layer can be created which could check other processes that whether processes are running properly[13]. If the process is not running properly that means process is under attack and to cure the damaged process kernel which is performing the functions of lymphocyte can kill, suspend or restart that process just as in human immune system. As each lymphocyte process could have a randomly-generated detector or set of detectors, living for a limited amount of time, after which it would be replaced by another lymphocyte. Therefore, it is impossible to attack the protection system because there would be no predefined location or control thread. If some lymphocytes are performing well and are useful for the system e.g. detecting new anomalies then the life span of these lymphocytes can be increased. Additionally, autoimmune responses e.g., false alarms could be prevented through a censoring process analogous to clonal deletion in the thymus.

System using lymphocytes has the ability to adapt to changes in user behavior and system software by changing lymphocytes. Different security levels could be adopted by increasing the life span of lymphocytes and number of detectors in the lymphocytes.

For implementation of these guidelines, an analog for peptide/MHC binding and a technique for eliminating self-reactive detectors is required[14].

Network protection of mutually trusting computers: Next guideline is to think of each computer as corresponding to an organ in an animal. Consider each process as a cell, but now a human being is a network of mutually trusting computers. Here the innate immune system is composed of host-based security mechanisms, combined with network security mechanisms and firewalls[15]. Kernel-assisted lymphocyte processes implement the adaptive immune system layer, with the one more characteristic that these lymphocytes could migrate between computers, making them mobile agents. Now one computer or a set of computers could then be reserved as a thymus for the network, which will select and propagate lymphocytes, each of which searches for a specific pattern of abnormal behavior. Centralized system is not required to coordinate response to security breach if these lymphocyte processes use negative detection. The detecting lymphocyte can take whatever action is necessary, possibly replicating and circulating itself to find similar problems on other hosts.

This guideline is similar to the previous one, difference is mobile detector processes or mobile agents are added here. Now it is able to detect the same class of anomalies. With the help of mobile agents anomalies detected on one computer could also be quickly eliminated from other computers in the network. It has similar requirements as before, except that it also depends upon a robust mobile agent framework.

Network protection of mutually trusting disposable computers: Now regard each computer as a cell, with a network of mutually-trusting computers being the individual. By default the normal defense the cell has is host-based security.. For computer the innate immune system consists of the network's defenses, such as Kerberos and firewalls. By creating a set of lymphocyte machines adaptive immune system layer can be implemented. Now the purpose of these machines is to monitor the state of other machines on the network. If any machine found infected, the problematic machine could be isolated perhaps by reconfiguring hubs and/or routers, rebooted, or shut down. But if the problematic machine were outside the network, a lymphocyte could stand in for the victimized machine, doing battle with the malicious host, potentially sacrificing itself for the good of the network.

These guidelines could address problems regarding compromised hosts, network flooding, denial-of-service attacks and even hardware failures[16]. However, this guideline is significantly more required than the previous two. An implementation of this guideline requires an MHC/peptide analog at the host level and should be based on a machine's network traffic, or based on the behavior of its kernel. To allow lymphocyte machines to isolate a given host a dynamically configurable network topology would be necessary. As used previous guideline, a thymus-type mechanism would be needed to prevent autoimmune responses[17]. An implementation would require that hosts must be somewhat interchangeable-otherwise the network could not afford the loss of any hosts.

CONCLUSION

Present world is the era of information technology which has made the sharing of information a click away. But this technology has generated adverse effects also one of which is virus i.e. with the generation of new technologies new viruses are also coming up every day. There are new anti-virus programs and techniques developed too. It is good to be aware of viruses and other malware and it is cheaper to protect your environment from them using latest antivirus software rather than being sorry. If your system starts behaving differently it means your system has been infected. This

infection can harm your computer in different ways like it may restrict some functions, delete files, format your disk and automatically shutdown your system. It is required to be little conscious of spyware and adware when you surf in the Internet and download files. Malware might be hidden in the files which looks interesting. A computer virus is a program that replicates itself and its motive is to spread out. Therefore by following above four guidelines which are based upon human immune system can be used to make more secure system from viruses. Not all viruses are harmful but some viruses might cause random damage to data files. There are many viruses which behave differently from general concepts regarding viruses e.g. Trojan horse virus and Macros. A Trojan horse is not a virus because it doesn't reproduce. The Trojan horses are usually masked so that they look interesting. These viruses that steal passwords and format hard disks. Macro viruses spread from applications which use macros. These viruses spreads fast through internet because people share so much data, email documents and use the Internet to get documents. Macros are also very easy to write. Some people want to experiment how to write viruses and test their programming talent. But they do not understand about the results for other people or they simply do not bother. The mission of viruses is to move from one program to other and this can happen via floppy disks, Internet FTP sites, newsgroups and via email attachments. Viruses are mostly written for PC-computers and DOS environments. Today every user has to deal with viruses.

For good security appropriate passwords, proper access controls and careful design are still needed. These protection measures act as similar as the body's skin and innate immune system, which are responsible for preventing most infections. This paper has focused on the human immune system's adaptive responses, because these are the types of mechanisms current computer systems do not have. By removing these shortcomings, it is possible to make computer systems much more secure.

REFERENCES

1. Wulf, W.A., C. Wang and D. Kienzle, 1995. A new model of security for distributed systems.Technical Report CS-95-34, University of Virginia.
2. Who writes malicious programs and why? http://www.viruslist.com/
3. Forrest, S., A. S. Perelson, L. Allen and R. Cherukuri, 1994. Self-nonself discrimination in a computer. In Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamos, CA, 1994. IEEE Computer Society Press.
4. Fred, C., 1987. Computer viruses. Computers & Security, 6: 22-35.
5. Blakley, R., 1997. The emperor's old armor. Proc. New Security Paradigms'96. ACM Press.
6. Forrest, S., A. Somayaji and D.H. Ackley, 1997. Building diverse computer systems. Sixth Workshop on Hot Topics in Operating Systems.
7. Computer Virus Classification. http://www.avp.ch
8. Kephart, J.O., A biologically inspired immune system for computers. R.A. Brooks and P. Maes, Eds. Artificial Life IV: Proc Fourth Intl. Workshop on the Synthesis and Simulation.
9. Hanshisals, M., Computer Viruses, Department of Computer Science, Helsinki University of Tecnology.
10. Tonegawa, S., 1983. Somatic generation of antibody diversity. Nature, 302: 575-581.
11. Inman, J.K., 1978. The antibody combining region: Speculations on the hypothesis of general multispecificity. In G.I. Bell, A.S. Perelson and Jr.G.H. Pimbley, Eds., Theoretical Immunology, pp: 243-278. M. Dekker, NY.
12. Forrest, S., A.S. Perelson, L. Allen and R. Cherukuri, 1994. Self-nonself discrimination in a computer. Proc. IEEE Symp. Research in Security and Privacy, Los Alamos, CA, 1994. IEEE Computer Society Press.
13. Inman, J.K., 1978. The antibody combining region: Speculations on the hypothesis of general multispecificity. In G.I. Bell, A.S. Perelson and Jr.G.H. Pimbley, Eds., Theoretical Immunology, pp: 243-278. M. Dekker, NY.
14. Somayaji, A., S. Forrest and S. Hofmeyr, Principles of a Computer Immune System. Department of Computer Science, University of New Mexico, Albuquerque.
15. Neuman, B.C. and T. Ts'o, 1994. Kerberos: An authentication service for computer networks. IEEE Commun. Mag., 32: 33-38.
16. Forrest, S., S. Hofmeyr, A. Somayaji and T. Longstaff, 1996. A sense of self for UNIX processes. Proc. IEEE Symposium on Computer Security and Privacy. IEEE Press.
17. Forrest, S., S. Hofmeyr, A. Somayaji and T. Longstaff, 1996. A sense of self for UNIX processes. Proc. IEEE Symp. Computer Security and Privacy, IEEE Press