WUJNS
Wuhan University Journal of Natural Sciences

Vol.10 No.1 2005 083-088

Article ID: 1007-1202(2005)01-0083-06

Mechanism and Defense on Malicious Code

□ WEN Wei-ping[1,2,3], QING Si-han[1,2,3†]

1. Institute of Software, the Chinese Academy of Sciences, Beijing 100080, China;
2. Engineering Research Center for Information Security Technology, the Chinese Academy of Sciences, Beijing 100080, China;
3. Graduate School of the Chinese Academy of Sciences, Beijing 100080, China

Abstract: With the explosive growth of network applications, the threat of the malicious code against network security becomes increasingly serious. In this paper we explore the mechanism of the malicious code by giving an attack model of the malicious code, and discuss the critical techniques of implementation and prevention against the malicious code. The remaining problems and emerging trends in this area are also addressed in the paper.

Key words: malicious code; attacking model; mechanism; defense; system security; network security

CLC number: TP 228.4

Received date: 2004-04-30

Foundation item: Supported by the National Natural Science Foundation of China (60083007) and the Special Funds for Major State Basic Research Program of China (973 Program) (G1999035810)

Biography: WEN Wei-ping(1976-), male, Ph.D candidate, research direction: theory and technology of network and information security and research on malicious code

† To whom correspondence should be addressed. E-mail: qsihan@ercist.iscas.ac.cn

0 Introduction

With the explosive development of the Internet, the threat of malicious code against network security and system security becomes increasingly serious[1].

Cohen pointed out the malicious codes' endangerments and the limitations for detecting them. He also drew the conclusion that the common detection methods against the computer viruses are undecidable[2]. The methods keeping the malicious codes away can be divided into two categories: detection methods and defense methods. The host-based detection methods mainly rely on the characteristics of the malicious code. For example, the characteristic string scanner method mainly relies on the code characteristics[2]; the behavior monitor method mainly relies on the behavior characteristics; in addition, the checksum method, the software simulation method and the VICE predictability method, etc, are often used[3]. The network-based detection methods are currently hot issues in the area of the malicious code counter-technology. These methods make use of the mass information and data coming from network, defend and detect the malicious codes existing in the large-scale network using the principles used in the Data Mining and the Intrusion Detection System. The defense methods can also be classified as two kinds: segmentation model and stream model[4]. The basic theory of the defense systems can be described as the following: A security system Based on some specified security strategies is formed and used to restrict or hold back the intrusion, infection and propagation of the malicious programs, accordingly control and even avoid the destructions caused by the malicious codes.

1 Relevant Conception

The computer viruses were the main form of the malicious codes early[5]. Cohen designed a destroying program that could replicate itself by running it[5] in 1980s. Adleman called it as a computer virus. It acted as the main part of the early malicious codes. Later, Adleman defined the malicious codes as the set of the programs that had the same properties. As long as the program has the characteristics of destroying, infection and simulation, it can be consider as a computer virus[6]. If all destroying programs are considered as computer viruses, the other important characteristics, such as the latency and the infection, etc, are easily ignored. In the late of the 1990s, the definition of the malicious codes has gradually become complexity with the development of the network. As the option of Grimes, the malicious codes are such programs or codes that propagate from the one system to another system through the storage medium and the network without being authorized and authenticated, and destroy the integrity of the computer system[1]. This definition includes the computer viruses, worms, Trojan Horses,Logic bombs, Bacteria, malicious scripts and malicious ActiveX, etc. From this definition, we can get two main characteristics of the malicious codes: unauthorized and malicious. We list the several main types of the malicious codes and their correlative specifications in the following Table 1.

Table 1 Several main types of the malicious codes

Type	Specification	Characteristic
Computer Virus	A set of computer instructions or program codes which can destroy the functions of the computer and the data stored in the computer, affect the normal uses of the computer, and are able to replicate itself.	Latent, infectious, destructive
Worm	The programs that can replicate itself through the network, consume the resources of the system and the network.	Attacking, propagation, diffuseness
Trojan Horse	The program codes that have the corresponding privileges, hide in the legal programs, do bad activities or access the system without authorization.	Cheating, convert, Remotely control
Logic Bomb	The programs that are inserted into the computer system and triggered with specific data or time, attempt to realize some destructive functions.	Latent, destructive
Bacteria	The programs that don't rely on the system software, are able to copy itself and propagate, take the consuming of the system resources as the goal.	Infectious, DosAttack Model

Although the behaviors and destructive degrees of the different malicious codes are various, their basic mechanisms are almost same. The whole attack process consists of six steps:

Step 1 Intruding the objects. It is essential to intrude its objects for one malicious code to implement various attack intention. Many ways can be used by the malicious codes to intrude the objects. For example, the programs downloaded from the Internet maybe include malicious codes. The E-mail we received may contain the malicious codes. When we install some software from the CD or floppy disk, our system is likely intruded by the malicious codes. Sometimes, some hackers or attackers maybe insert the malicious codes into their objective systems in advance.

Step 2 Keeping or elevating the current privileges. In order to accomplish their propagation and destruction, the malicious codes try their best to steal the users' or process's legal privilege.

Step 3 Concealing itself. In order to prevent itself from being detected, the malicious code usually take some methods to conceal itself, such as rename or delete the source files, or modify the security strategies in the target systems.

Step 4 Latency. When the malicious code intrudes into one system, it doesn't destruct the system, immediately, until the conditions the malicious code need are ready and the privileges are enough.

Step 5 Destruction. Destruction is one of the essences the malicious codes hold. The ultimate purposes of malware is to make the information lost or disclosed, or destroy the system's integrity, etc.

Step 6 Repeat all the above steps to attack the new objects. The malicious codes' attacking model is illustrated by Fig. 1. The attacking process of the malicious codes can be composed of several parts of or the whole model. For example, the computer virus's activities mainly include step 1, 4, 5, 6. The network worms' activities mainly include step 1,2,5,6. The Trojan Horses' activities can be described with step 1, 2, 3, 5. The Logic bombs can be described with step 1,4,5. The Bacteria mainly include step 1,5,6. Other malicious codes' activities can also be mapped into relative parts of the model illustrated by Fig. 1. Among all the steps, step 1 and step 5 are necessary.

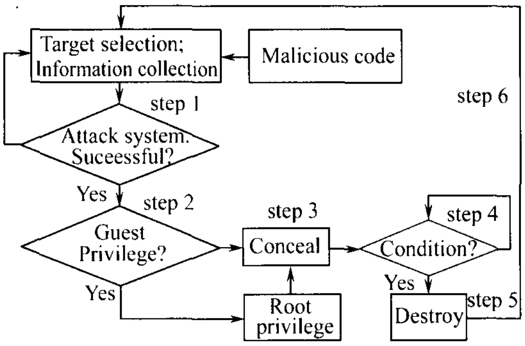

Fig. 1 Attack model of malware

2 Implementation

2.1 Anti-Debug

With the anti-debug technologies, the malicious code can improve its camouflage ability and prevent itself from being easily decoding, as well as increase the difficulty of detecting and clearing up the malicious codes. The anti-debug technology used currently usually belongs to one of the two categories: anti-dynamic-debug technologies and anti-static-debug technologies.

2.2 Polymorph

Using the polymorph techniques, the malicious codes insert the objective program various codes when it infect one object. Nowadays, the polymorph techniques mainly include the following types:

1) Reassemble technology. Mutation Engine anti-assembles the binary, decodes every instruction, computes the length of every instruction, and replaces the instruction with the same function. "Regswap" used the register-switch technology.

2) Shrinking technology. The polymorph tool scans all the instructions of the malicious code and shrinks the instructions that can be shrank. Shrinking technology does not change execution intention of code. Because the Shrinking changes the length of the malware body, the jump instructions need to be relocated.

3) Expandability technology. This technology expands the instructions with other instructions with the same functions. This technology is contrary to the compression technology. The space that the expandability technology can modify is by far more than the compression technology can do. Some instructions maybe have several ten, or even more than 100, ways to replace. Because the expandability technology also needs to change the length of the instructions, the jump instructions need to be relocated, too.

4) Junk code technology. This method mainly inserts the junk codes (for example, the null instructions, jump next instruction, etc) into the source code.

5) Recompiling technology. The malicious code which uses the repeated compiling technology embeds the source code of the virus. It needs to take the compiler by itself or use the compiler that the OS provide to recompile the source code. This technology not only achieves the goal of polymorph, but also makes the foundation for the emergence of the multi-flat viruses. Especially the various LINUX/UNIX OS's default configuration usually include the standard C language compiler. The macro viruses and the scripts viruses are the typical examples which introduce the repeated compiling technology.

Polymorph is an important technology to strengthen survivability of the malicious codes, and is also the hot and puzzle problem. In our opinions, the developing trends of polymorph can be described in several aspects. Firstly, the polymorph will appear in the other malicious codes besides the viruses. Secondly, polymorph will not only work on the virus body but also distort the other parts or the whole that the malicious codes adhere to. This increases the difficulty in cleaning the malicious codes. The third is that the function of the polymorph engine will be more powerful. The polymorph engine trends to use the construction insertion technology. Lastly, the malicious codes which adapt to the multiple OS platforms and the multiple hardware platforms will appear in the gross.

2.3 Tri-Thread

The Windows OS introduces the conception of the thread. One process can hold several concurrent threads. The tri-thread technique means that one malicious code's process initiate simultaneously three threads, among which one acts as the main thread which takes charge of the remote controls, one is the monitoring thread which takes charge of checking if the malicious codes have been deleted or have been forbidden to run by itself, one is the daemon thread which is inserted into the other binary files and keep synchronization with the malicious codes. Once the malicious process is stopped, the daemon thread will resume the process and provide the main thread with the necessary data to guarantee the durable running.

2.4 Process Injection

Current Operation Systems all have the system services and the network services. These services will be automatically loaded as soon as the system startups. With the process injection technologies, the malicious code inverts itself into the program codes relative with the system services and the network services. The malicious code takes these program codes as its carriers to realize its hiding and automatic running. Once this kind of the malicious codes is installed, it will be automatically loaded into the executable file's process and be loaded by several services in the later. The malicious code always remains active during the system's running because the services are not over until the system is shut down.

2.5 Repeated Use of the Port

With this technology, the malicious codes can use the ports (for example, 25,80, 135 and 139) opened already to transmit the data, which not only can avoid the detection of the firewall, but also decrease the number of the new ports. The repeated use of the port must assure the normal running of the default services and has very good effect in cheating the detection devices.

2.6 Port Reverse-Connection

The firewalls usually have strict access control strategies for the communications from the outside to the inside. But for the communication from inside to outside, the firewall gives less strict access control. The port reverse-connection techniques means that the server attacked by the malicious code (The system that has been controlled) actively connects the client of the malicious code (The system that is used to control the objects).

2.7 Automatically Designing

The automatically designing technology is opposite to the manual analysis technology. "The Computer Virus Designer" makes it possible for the users who understand nothing about the computer viruses to generate the computer viruses which are various in the algorithms and the functions. "Polymorphism Generator" can compile the common viruses to form the complicated and changeful polymorphism viruses. The polymorph engine can change the program codes while keeping the same function. The "Dark Avenger"[7] is a remarkable example.

3 Defense

3.1 Characteristic-Based Scanning

The characteristic-based scanning technique, which is the most common used to detect the malicious codes, is based on the mode matching theory[8]. Before the scanner works, the file about the characters of the malicious codes must be ready. And then according to the characteristic string in the file, the scanner tries to find the string that matches the characteristic string. User can update the scanning software by updating the characteristic file to find the last-version malicious code. This technology is widely used by the current anti-virus engine. Fig. 2 illustrates its workflow.

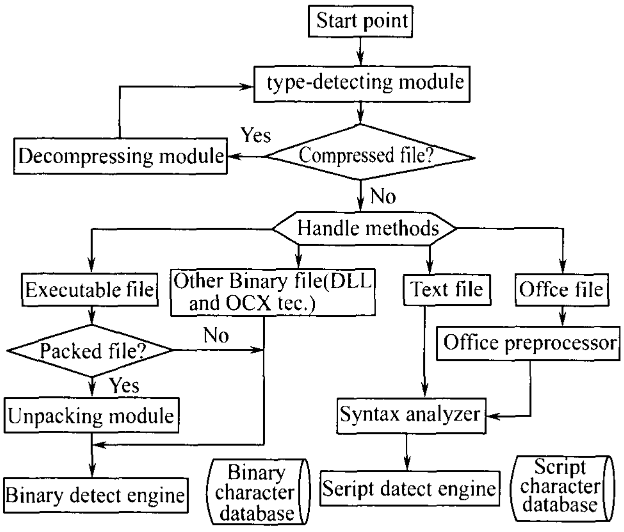

Fig. 2 Work flow of common anti-virus engine

Currently, the character-based scanning technologies have two main problems: 1) It is an algorithm based on the characters of the malicious codes, so that does not well work against the encoded, distorted and the unknown malicious codes. 2) The user needs to constantly update the detection engine and the characteristic database. It can't pre-alarm the intrusion of the malicious codes, and can only work after the intrusion has happened.

3.2 Sandbox

This technology creates a "Sandbox" for each application based on the resources the application need to access and the privileges that the system authorize to confine the running of the malicious codes. Every program runs in its own protected "Sandbox", and will not affect the operation of the other programs. Similarly, these programs can't also affect the normal function of the operation system. The system and driver programs also work in their own "Sandbox". The Berkeley lab of the California University developed the Sandbox system based on the Solaris[9]. In this system, the application runs by calling the system's low-layer functions. The system automatically judges if the low-layer functions that the application programs invoke accord with the system security strategies, and determine whether the program is permitted to run. Sandbox provides a configuration file for each application to confine the resources that the application can access and the privileges that the system assigns it. Windows XP provides one strategy for the software confinement to isolate the codes that have latent menaces[10], which in fact is a kind of the Sandbox. It can prevent the system from the infractions that are caused by the malicious codes coming from the Email or Internet. These strategies permit the user to choose the ways that the user manages the application programs from the Confining Running and the Forbidding Running. Running the untrusted codes and scripts in the Sandbox can confine or even prevent the destruction of the malicious codes to the system's integrity.

3.3 Secure OS

One important step that the malicious code successfully intrudes a system is to get the control privilege of the system and have the system distribute enough resources to it. Without enough privileges, it is impossible for the malicious code to accomplish its goal, or it at best can accomplish part of the goal. If the operations that the program accesses the objects of the system can be reasonably controlled, the damages to the system will be confined [11]. With the mandatory access control policy of the secure Operation System, the space of the computer system can be divided into 3 parts: the system management space, the user space and the protection space. The mandatory access control strategy divides the users entering the system into two kinds: the common users who have no privileges and the system manager. The common user can't read or write the system management space, such as the TCB, the audit file, etc. The TCB protects itself with the access-isolated method and makes sure that it is trusted. The user space contains the user's application programs and data, and can be accessed by the user. The protected space contains the programs and the data that can't be modified by the procedures in the user space, but can be read by users. Generally, the common commands and programs are stored in the protected space to facilitate the users to use. The common users can only read the programs and the data in the space, which constraints the propagation of the malicious codes. In the user space, because the user's secure level is different, even though the malicious codes breakout, they can only infect the programs and data of the users with the same secure level, which reduce the propagation scope.

Due to the limit space in paper, the other defense techniques against the malicious codes, such as the activity monitoring techniques, the bacteria techniques, the dynamic trap techniques, the simulation of software techniques and the VICE-Virus Instruction Code Emulation-techniques, etc. can be found in Ref. [11].

4 Research Trend

The malicious codes develop form the original single infection and single activity to infecting dependent on the network and combining the E-MAIL, file transmission and other infection ways and synthesizing the hacker and virus and other many attack methods together. In our opinion, the development trends of the malicious codes should be:

1) The network will be the main approach that the malicious codes propagate. The network worms will become the most destructive type of the malicious codes.

2) The Trojan Horses will exist anywhere. They make use of the secure leaks of the OS and the network to lurk and will be the shadiest malicious codes.

3) The combination of the hacker techniques and the virus techniques will generate the new generation of the malicious codes that propagate actively.

4) The manners that the malicious codes propagate will be diversity.

5) The cross-OS malicious codes, web page malicious codes, malicious scripts, cell malicious codes and the malicious codes against the electrical appliances will emerge and propagate.

From the malicious code's development, we can know that the new generation of the malicious codes will take on the highly activity in the network surrounding. Due to the distribution of the malicious codes source, the conventional detection and the defense with the host-based virus defense techniques and the virus firewall techniques will not be applicable to the future malicious codes [12]. Thus, we think that making good use of the mass information and data provided by the network and the distribution architecture to realize the detection and defense of the large-scale network surrounding will be the new direction against the malicious codes. We have done some researches on confronting the malicious codes on the large-scale network surrounding. The relative results will be addressed in the other papers.

5 Conclusion

The detection and defense of the malicious codes is a long-term procedure because:

1) The type of the malicious codes is various and the forms are complexity.

2) The mechanism of the intrusion and infection and the trigger differs in thousands ways.

3) It is difficult to precisely predict the new malicious codes. Therefore, we must not only grasp the current execute mechanism of malicious codes, but also strengthen the researches on the development trends, and do actually prevent accidents before they break out.

References

[1] Grimes R A. *Malicious Mobile Code, Virus Protection for Windows*. Grarenstein: O'Reilly Press, 2001. 1-2.

[2] Cohen F. *A Short Course on Computer Viruses*. Pittsburgh, PA: ASP Press, 1990.

[3] Qing Si-han, Liu Hai-feng, Liu Wen-qing. *Introduction to Operation System Security*. Beijing: Science Press, 2003. 108-113(Ch).

[4] Yang T. *SUNIX Secure Operation System* [Ph. D thesis]. Beijing: Institute of Software Chinese Academy of Sciences, 1993.

[5] Cohen F. *Computer Viruses* [Ph. D thesis]. Los Angeles: University of Southern California, 1985.

[6] Tian C, Zhen S R. Computational Model of Computer Virus. *Chinese Journal of Computers*, 2001, **24**(2): 158-163.

[7] Sarah G. Inside the Mind of Dark Avenger. *Virus News International*, 1993, **20**(1): 48-50.

[8] Grenander U. *General Pattern Theory: a Mathematical Study of Regular Structures*. Oxford: Clarendon Press, 1993.

[9] Goldberg I, Wagner D, Thomas R, *et al*. A Secure Environment for Untrusted Helper Applications: Confining the Wily Hacker. *Proceedings of the* 1996 *Usenix Security Symposium. San Jose, CA. July* 22-25, 1996. *http://citeseer.ist.psu.edu/goldberg96secure.html.*

[10] Lambert J. Soft Restriction Policies in Windows XP. *Proceedings of Virus Bulletin Conference, Hyatt Regency*, New Orleans, LA, USA. September 26-27, 2002. *http://www.virusbtn.com/files/johnlambert_vb2002.pdf.*

[11] Wagner D A. *Janus: An Approach for Confinement of Untrusted Applications* [Master thesis]. Berkeley: Computer Science Division, University of California, 1999.

[12] Understanding Symantec's Anti-virus Strategy for Internet Gateways. May 1999. *http://www.symantec.com/avcenter/refe rence/wpnavieg.pdf.*

□