Research and Application of Malicious Code Detecting Platform in Intranet

Xiaojun Zhang, Yong Sun, Xuefan He
Beijing Aerospace Control Center, Beijing 100094, China;
zxjyn2006@126.com

Keywords: Malicious code, dynamic analysis, trojan horse, vulnerability.

Abstract. Intranet and extranet information interactions are increased constantly aiming at IP network equipment interconnection, which become actual conditions of more and more complex uncertain systems. A malicious code behavior detecting platform is designed and realized by combining with the characteristics of network attack, attack methods and principles used by the malicious code. Automatic flow of analyzing malicious codes is realized. It is preliminarily applied in internal IP network. Malicious code detecting platform provides strong guarantee and support for stable, efficient and safe operation of space detection task networks, which provides reference for maintenance personnel to implement emergency disposal quickly and properly.

1. Introduction

Independent sky-earth integration network formation protocol system and application of highly efficient information transmission technology will realize heterogeneous network interconnection and automatic storage distribution of spatial information with development of space detection tasks. IP network, as a foundational support platform, is responsible for convergence and distribution of all data. Although IP network belongs to a physical isolation independent network, it becomes a more and more complex uncertain system due to equipment interconnection, and constantly increased interaction among system, extranet and international network. In the process of information exchange, the system is still under various threats due to long transmission distance, uncontrollable region, excessive access units and other reasons. People gradually realize that network security can not be completely realized by only relying on safety protection software with the advance of computer technology in view of diversified information attack means [1]. The malicious code detecting platform based on cloud safety is the foundation for emergency response of the information security system and the computer forensics. Detailed information is provided for system recovery and loss assessment by analyzing the function and purpose of the malicious code. Malicious code is also developing with the development of network technology. When the aggression is increasing, self-concealment of modern malicious code is also improved generally by polymorphism, deformation, anti-debugging and other means. In addition, malicious attacks also become easy and simple with the emergence of some malicious code automatic generation tools and attack codes; malicious code behavior detection is becoming a hot spot of research [1].

2. Design of Network Malicious Code Detecting Platform Based on Cloud Safety

The malicious code detecting platform is mainly used for detecting and analyzing all data that might be used for malicious attacks [2]. A variety of detecting methods are used, which mainly include static format parsing and dynamic execution to detect malicious sample files. Virtual load execution and dynamic monitoring are utilized for in-depth safety analysis, thereby effectively detecting 0day format overflow in response to senior safety threat, executable sample behaviors are deeply extracted, and detailed report file structure and behavior report are provided.

Multi-level malicious code intelligence analysis detection technology is adopted in the platform, which is mainly divided into three levels of detection mechanisms. The first level refers to detection based on virus feature library, which consists of a large number of black and white lists and can

effectively improve the detection efficiency. The second level refers to static detection, including format recognition parsing, character string information extraction, vulnerability detection and other means. Files are further analyzed. The third layer refers to dynamic detection, including process operation, file operation, registry operation, network communication protocol access, network data URL and other detection points. In-depth detection behavior triggering is executed aiming at files. Multi-level detection mechanisms ensures the detection performance of the whole model, therefore its detection speed, detection accuracy and detection efficiency are much better than ordinary malicious detection system.

Network malicious code detecting platform is used for analyzing the behavior characteristics of files, and collecting suspected malicious code samples from the host. Data of multiple data records points is aggregated. The same detected object is detected locally and in the detection center by many different detection methods. It is distributed rapidly through internal cloud distribution mechanisms. The problem of difficult intranet upgrading is solved to certain degree. Automatic flow of analyzing malicious codes can be realized by the platform, the influence of sample cod on operating system files, networks, registries, processes, etc. is recorded. Detailed attack information of the malicious code is collected. Attack technology for information concealment and attack technology for process concealment in the registers are detected. Unknown malicious code is detected. The workflow is shown in figure 1.

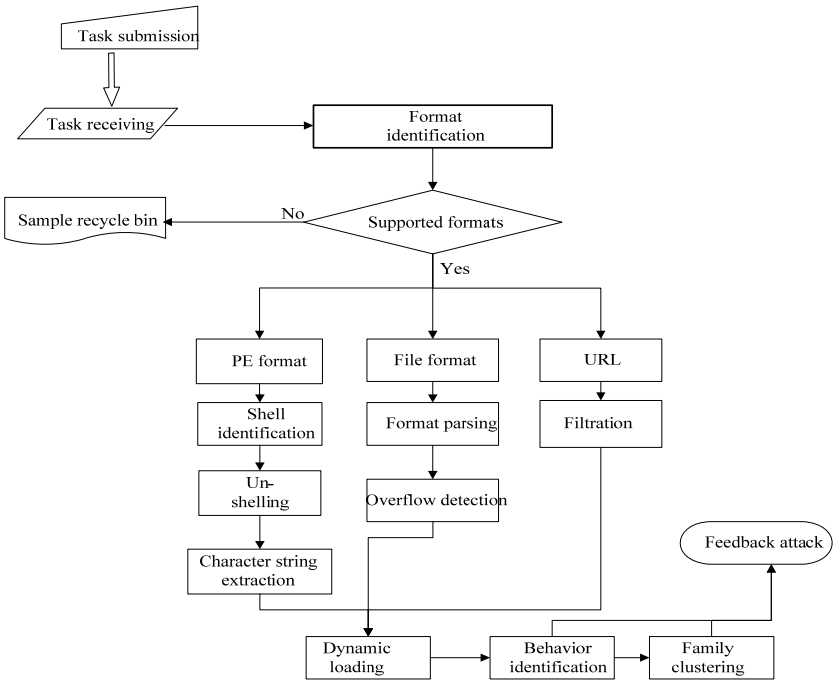

Fig.1 Workflow chart of malicious code detecting platform

3. Malicious Code Detection Based on Kernel Driver

Dynamic analysis refers to an analysis method of comprehending functions of malicious codes by monitoring the execution of the malicious code. Technical people adopt code-injection technique, etc. to analyze the malicious codes with the increase of malicious code attacks. But there are defects more or less [3]. For example, creation of remote thread for execution of binary code can lead to crash of target process.

Dynamic detection of malicious code based on kernel driver is operated in the system kernel with the drive method. The execution program in the dynamic monitoring system does not affect the performance of the system, and attack information is timely reported to users, thereby enhancing the system's overall safety. One kernel driver is formulated, suspicious injection behavior in each process of the system is detected dynamically, and the system calling and API function application condition in the malicious code execution process are monitored for analyzing the malicious code. Internal

kernel Hook technique is adopted for key system calling. Ring3 and Ring0 driver combination form is mainly adopted for dynamic detection, thereby forming a three-dimensional monitoring system.

Ring3 monitoring is mainly based on remote injection mode. Inline hook is carried out on API of system calling. The remote thread injection method not only can be injected into the startup process, but also can be injected into a newly-established process. Different opportunities can be selected for injection during new establishment of process, such as Hook NtResume Thread can realize global Hook, while Cuckoobox can be injected into new start-up process in the function of Create Remote Thread. In addition, the detection program can resist modification on \Device\Physical Memory in order to protect the system service table. A protection module is developed for warning users when kernel drivers are added in the system in order to prevent malicious codes from detection avoidance by the detection method.

Malicious code detection program based on kernel driver is mainly composed of three parts mainly for discovering kernel driver load and protection system service table, namely it is used for protecting the detection program itself. The other part is responsible for malicious code injection detection. Figure 2 is a malicious code dynamic detection program structure based on kernel driver. Suspicious list and dangerous list are used for storing process handle and base address. When the detection program is loaded into memory, five system calls are hooked, including ZwWriteVirtualMemory, ZwAllocateVirtualMemory, ZwOpenSection, ZwCreateThread and ZwLoadDriver. The system calls are called by a process. These requests can be interrupted by the detection program, and corresponding system parameters are checked.

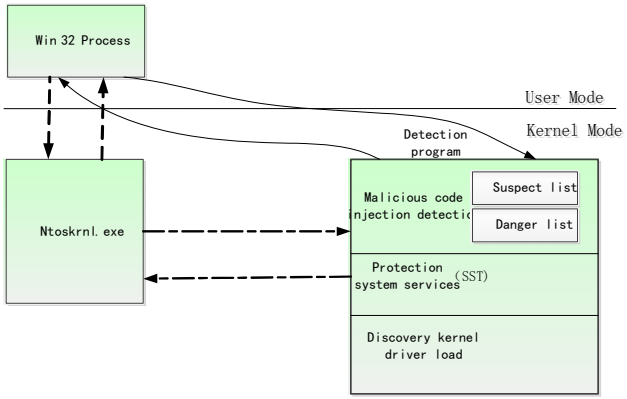

Fig. 2 Malicious code dynamic detection program structure based on kernel driver

Detection of malicious code injection behavior is composed of two parts. One part is used for judging whether the injection belongs to operation among processes or not, and storing the handle and base address of the process into the suspicious and dangerous lists. The information in these two lists will be used for detecting whether a malicious code is injected or not. Several key API functions must be called in one malicious process for finishing injection. Several key API functions include WriteProcessMemory, CreateRemoteThread and VirtualAllocEx in order to monitor the entire operating system, and prevent malicious process from directly calling system calls corresponding to these API functions. Hooks are set for these system calls for monitoring the injection behaviors of the malicious code. Hooks are set for the three system calls to monitor the processes operated in the system, including ZwWriteVirtualMemory, ZwAllocateVirtualMemory and ZwCreateThread. Meanwhile, the entry addresses of the three system calls are directed to our functions through modifying the protection system service table. DLL injection and binary are identified in the region clearly through detecting parameters of these system calls.

4. Key Techniques of Malicious Code Detection

Malicious code detection and defense model involves the following main key technologies [4]: vulnerability identification technology, unknown Trojan horse behavior identification technology, Trojan horse family varieties identification technology, intelligent learning, self-protection ability based on the sandbox, sandbox monitoring mechanism, etc.

4.1 Vulnerability Identification Technology

When vulnerability overflow files are utilized in static inspiration identification, a piece of code is used in both unknown vulnerabilities and known vulnerabilities, which is jumped to executable codes which are carefully constructed by attackers, and the code is Shellcode [5], permission can be obtained or other malicious attacks are carried out. Some Shellcode characteristics can be summarized for detection analysis. The overflow files are statically identified through recognition of Shellcode. The original file formats should be properly transformed for Shellcode detection, including identification of original binary, hex hexadecimal and Base64 data, and they are handled uniformly.

Unknown vulnerability identification technologies are monitored dynamically. Since the attacks are transmitted by overflows in many formats, analysis and monitoring in the aspect of static overflow files are monitored, including common formats of PDF, MS word, XLS, PPT, RTF, WPS, etc. The system action behaviors generated by malicious format overflow are monitored. The load process of files can be monitored on the one hand, all process activities in the system can be monitored comprehensively on the other hand. The simulation double-click starting method of the starting program is exclusive in the system. Since overflow program depends on environment frequently, the method of simulating double-click for opening the resource manager is adopted aiming at the successful overflow under the double-click execution condition due to over-flow failure of calling API startup by other program. Data of different angles are discovered for comprehensive analysis and mutual complementation, thereby obtaining sample monitoring behaviors and data more comprehensively, and the dynamic monitoring diagram is shown in figure 3.

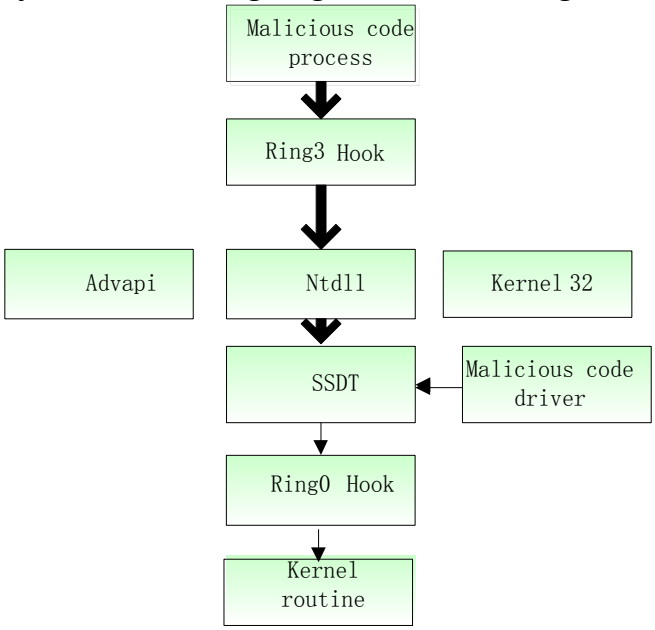

Fig. 3 Dynamic monitoring diagram

4.2 Trojan Family Variant Recognition Technology

The vector of each sample is constructed according to existing dynamic and static information of sample extraction, including file formats, character string information, shell, compilers, version, API, etc. Cluster includes hierarchical clustering and fast clustering LSH algorithms with similar vectors in pairwise comparison. Some known family variants can be classified. Unknown variant samples are newly clustered. The generated center point can be regarded as the quick comparison node of classification after clustering by K center point method. If the similarity of new samples and any existing center point is higher than certain threshold, it is judged as an existing family. The comparison process can be accelerated by LSH algorithm. The sample classification process diagram is shown in figure 4.

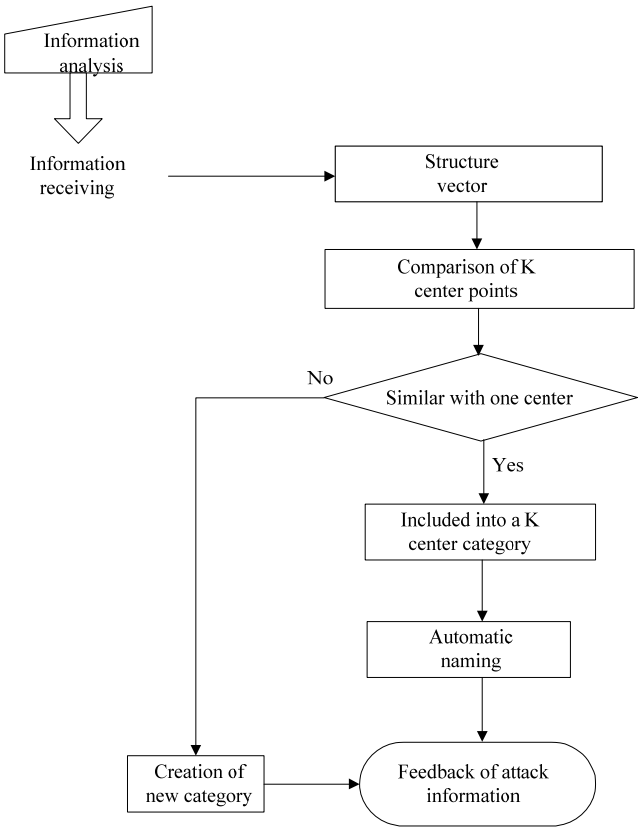

Fig.4 New sample classification process diagram

5. Detection Result Visualization

Task IP network does not support the active detection to obtain network topology structure from the perspective of safety, thereby it is proposed that the network topology relationship is stored in the form of data table. The network topology structure is visually displayed through a static mode. The network topology structure is visualized by the hierarchical structure according to the network host data table, and the hierarchical network topology design is shown in figure 5. The abnormal detection results are visualized in charts, graphs and other modes in the visual image of network topology structure. The visualization results show the network nodes or links with abnormities, and related abnormity information.

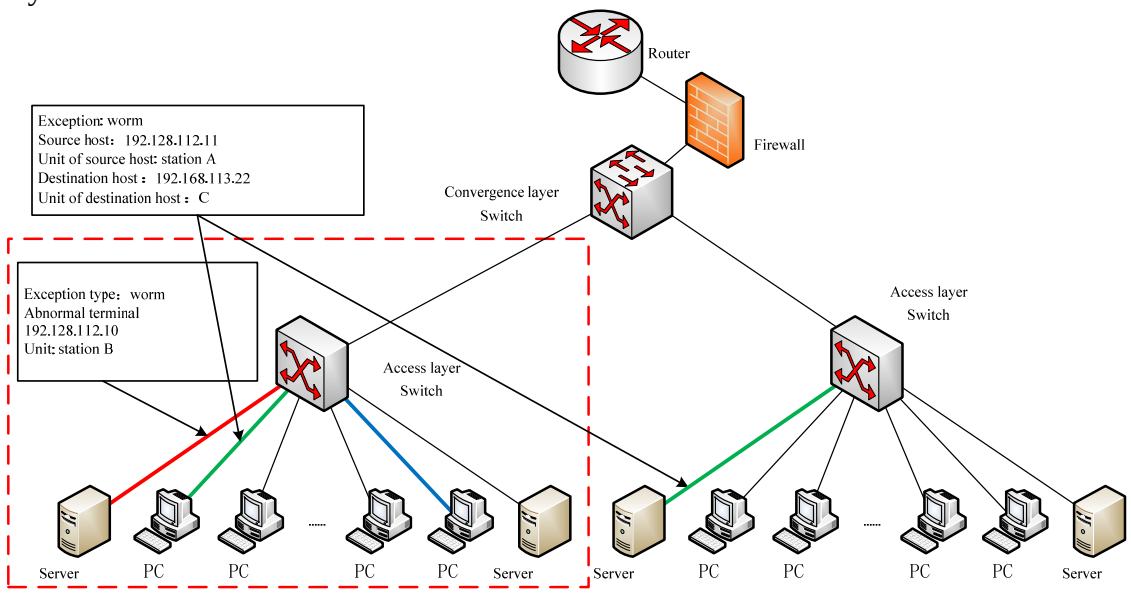

Fig. 5 Abnormity detection result virtualization effect picture

6. Conclusion

In the paper, the malicious code detecting platform based on cloud safety intranet is studied and applied. A multi-level malicious code intelligent analysis detection technology is adopted for realizing automatic flow of analyzing IP network malicious codes. Network safety is an eternal topic. It is necessary to take a proactive mode to realize active defense. Safety incidents are analyzed rapidly in real time; thereby there is a long way to go for guaranteeing system safety away from worry.

References

[1]. Wei Weimin, Yuan Zhongxiong. Research and Practice of Network Attacks and Defense Techniques. Netinfo Security, 2012,(12): 53~56.
[2]. Zhang Qunling. Design and Test of Malicious Code Detection system. Beijing: Beijing University of Posts and Telecommunications, 2012.
[3]. Li Wei, Su Purui. Dynamic Detection Technique of Malicious Code Based on Kernel Driver. Journal of Graduate University of the Chinese Academy of Sciences, Sep. 2010. 6-7
[4]. Zhang Chi. Quantitative Evaluation Algorithm of Network Security Situation. Henan Science, 2013, 31(7):985~987.
[5]. Zhang Erchao. System Security Technology Research. Beijing: Beijing University of Posts and Telecommunications, 2013.