A Sandbox Designed on User-level Virtualization Platform

XIE Jin
Information science and Engineering
Hunan University
Chang Sha, China
357424562@qq.com

***Abstract*—Network has been widely applied in all aspects of life with time. Spread of malicious programs and harm thereof are also gradually increased with network. Sandbox provides high isolation environment for operation of suspicious program, thereby detecting malicious code effectively. However, there are some problems and disadvantages in sandboxes which are popular at present. Therefore, we establish a sandbox on user-level virtualization platform, which is called Dune[1]. The so-called user-level virtualization refers to a virtualization platform capable for providing direct and safe privileged operation for application programs. It is called Dune[1]. Compared with VMM [2] which provides support for operation system, Dune is more compact and lightweight. Meanwhile, sandbox, as an application program, is operated under dune, which can be operated under privileged mode by the aid of VT-x[3]. Privileged operation can be provided directly and safely, mode switch can be reduced compared with sandbox in the application layer in the aspect of intercepting API calls[4]. Experimental results showed that dune-based sandbox can guarantee higher performance on the basis of smaller scale.**

Keywords—sandbox, dune, hardware support, API interception, information security

I. INTRODUCTION

In recent years, network application is gradually popularized and deepened aiming at both individuals or enterprises with the popularity of computer and rapid development of network technology. Information safety has become an indispensable research field of modern technology development with rapid development of Internet information technology. Everything has two aspects. Explosive development of information science is also provided with negative influence which can not be ignored[5]. The destructive force and infection of malicious code can not be ignored firstly. The malicious programs not only can reduce ability and safety of computer system, but also violates personal and corporate data privacy, thereby resulting in huge economic losses.

Therefore, sandbox technology [6]is produced as a result. Sandbox generally refers to an accurately controlled security platform, and it will monitor each system calls. When resources are applied and tampered by malicious programs, the phenomena can be effectively detected by sandbox, and then the harmful behavior of malicious programs on actual system can be prevented through redirection technology.

Virtualization technology has a main function of isolation in information security category [7,8]. Hardware supported virtualization technology is applied into the sandbox, thereby effectively improving the performance of sandbox.

Main content of the thesis:

The first chapter: introduction, research background and significance of the thesis were proposed, and two important aspects involved in the thesis were introduced.

The second chapter: The basic principle and implementation mechanism of Dune were introduced.

The third chapter: Principles of our sandbox, features and advantages thereof were realized on the dune module.

The fourth chapter: experiment, it included introduction of experiment environment, realization of several functions in sandbox, and performance test.

The fifth chapter: conclusion. All contents in the thesis were summarized simply. Meanwhile, problems and disadvantages of sandbox were analyzed.

II. VIRTUALIZATION TECHNOLOGY

Virtualization technology of Hardware Abstraction Layer (HAL) was nearly completed in IBM System Generation 1 to Generation 4 back in the 1970s. Loopholes and defects on system architecture were filled by virtualization technology and further innovation. HAL virtual machine also have gained increasingly profound development and attention in the aspect of efficient isolation with more mature and perfected technology as well as increasing demand for virtualization.

However, most CPUs (such as Intel X86) cannot perfectly support virtualization. A part of privileged instructions cannot actively take from low privilege level into kernel privilege level, thereby virtual machine monitor cannot intercept the privileged instruction calls.

Now, the above problems can be solved through three methods:

Full virtualization technology [9] paravirtualization technology[10] and hardware support technology based on dynamic instruction transformation.

An intermediary layer was formed between the virtual machine and hardware at the bottom layer operated under the virtual machine in full virtualization through virtual machine management program. Instructions of processor are intercepted by virtual machine management program, and it is coordinator for instructions to use hardware resources and dispatch peripherals. Virtualization can be supported in paravirtualization through modifying kernel of client operating system. Instructions of the above processor are directly changed by hardware SUPPORT virtualization . Virtualization can be supported by semantics itself. Intel Company launched VT-i and VT-x, which belong to hardware support virtualization technologies. New processor mode is added, which is called VMX , and virtualization is supported through mode switching.

A. Introduction of Dune

Dune is a virtual platform, which is established on the basis of hardware support (VT-x) and allows user program to use system privilege directly and safely. Application program operated on dune can safely adopt abnormality and virtual memory, which can be operated under privileged mode, etc. as shown in table 1.

1. TABLE I. SYSTEM PRIVILEGE

Mechanism	Privileged Instructions
Exceptions	LIDT,LTR,LRET,STI
Virtual Memory	MOV,INVLPG,INVPCID
Privilege Modes	SYSRET,SYSEXIT,IRET
Segmentation	LGDT,LLDT

Meanwhile, dune can be normally used without changing system kernel as a Linux loadable module.

Therefore, our user-level virtualization is safe isolation environment provided here aiming at application program as its name implies. System privileges can be safely used by application program through dune. It is not as complicated as full virtualization compared with two previous virtualization technologies, and such technology is different from paravirtualization which changes operating system kernel.

B. System Structure of Dune

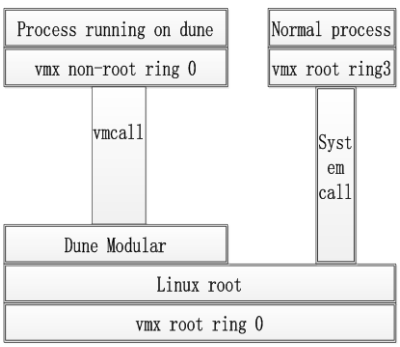

Fig.1. Dune System Structure Drawing

It is a system schematic diagram of application program which is operated in our dune platform environment. Our kernel modules are located at the bottom. It is operated under privilege level 0 mode under VMX root mode provided by VT - x, namely kernel mode. Dune can be loaded on kernel as a loadable module. The right side of the drawing displays that common program is operated under VMX root ring 3 mode, namely user mode. When system resources are used by application program under he mode, the kernel can be called through the system, and related requests can be completed by kernel. The application program is different in that application program operated in dune module can be operated under VMX non-root mode. When system resources are used by the program, the kernel is not needed for direct and safe use. Dune also provided support on original system call. When kernel is needed by the application program operated under dune mode, VMCALL can be used for meeting the purpose. In summary, dune, only provides operation environment of a program rather than supporting operating system compared with VMM virtual machine. Therefore, it is more compact and simple.

III. SANDBOX

A. Classification of Sandboxes

Sandboxes can be divided into three categories according to running environment: (1) sandbox on the application layer; (2) sandbox on the kernel layer; (3) mixed sandbox.

The main program of sandbox on the application layer is operated on the system application layer. It has advantages of simple implementation and convenient deployment. It has disadvantages of lower isolation performance, and complete reliance on kernel safety mechanism of operation system. In addition, once the operating system vulnerabilities are found and utilized by malicious program. Sandbox on the application layer can not exert own role completely with worse performance. Chrome browser belongs to such sandbox.

The main program of the sandbox on the kernel layer is operated on system kernel layer, therefore it has the same privileges as the kernel. Hardware protection mechanisms can be freely used for building strictly isolated security environment. In addition, since function code is realized in the kernel layer, switching frequency of user layer and kernel layer in the monitoring process can be avoided, thereby ensuring basic performance of user related program. Sandbox on the kernel layer has disadvantage of difficult development. In addition, since the sandbox has the same safety privilege level as the system kernel, the whole system can be endangered once the sandbox itself has defect. DD/OS (Device Driver OS) is a more typical sandbox on the kernel layer.

Our sandbox can operate privileged mode VMX non-root mode through dune module, which belongs to sandbox on the kernel layer. Therefore, higher performance can be kept during complete isolation of suspicious program. Sandbox structure drawing is shown as follows:

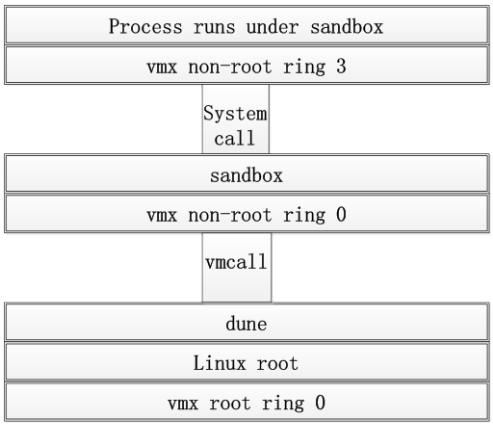

Fig.2. Sandbox operated on dune module

B. *Api Hooking*

API hooking is also called API hook, and it can be used for detecting and intercepting system calls. Application program can use or modify system resources through API functions. Supervision on API function is the important function for realizing sandbox safety isolation environment.

Since our sandbox is operated under non-root kernel mode, when any system call issued by application program is detected, we change the system call starting address in the kernel address, and it can be transferred to function inside the sandbox, namely hook function dune_syscall_handler. Therefore, API hooking can be realized.

C. *Realization of Sandbox*

Sandbox has the following main functions: (1) file redirection, file system isolation; (2) process isolation; (3) monitoring of program behavior;

File redirection: this is the most basic and most important function of sandbox, application program can modify the system mostly through file modification. We copy the file to a specific place when system file modification by application program is detected by sandbox. It is called shadow file. Then, modification of application program on system file is completely transferred into shadow file. The original system file can be changed during normal retreat of sandbox. Once suspicious behavior of program is discovered during the period, we can achieve safe rollback of the system directly through discarding shadow file.

Process isolation: the program operated in the sandbox can call new process through system calling (such as sys_execev sys_fork). We should achieve the follows: the new process can be operated in safe environment of sandbox when new program called by application program is discovered. Monitoring of program behavior: program behavior is based on system resources called by it. The sandbox can record all system calls of the program through API hook, thereby monitoring behaviors thereof.

IV. EXPERIMENT

A. *Experiment Platform*

Experiment platform of the thesis refers to a 64-bit Linux operating system. It is required that VT-x Intel CPU should be supported by hardware.

B. *Concrete Realization of Sandbox*

File system isolation: it belongs to related system calls of file system, such as open, chmode, create, chown, mkdir, etc. Calls about file system are undoubtedly the most unsafe. All files of systems can be changed after the highest authority is obtained by the program. System kernel files can be changed through reading and writing, and system can be easily failed. Therefore it must be monitored. Monitoring drawing of sys_open in sandbox is shown as follows:

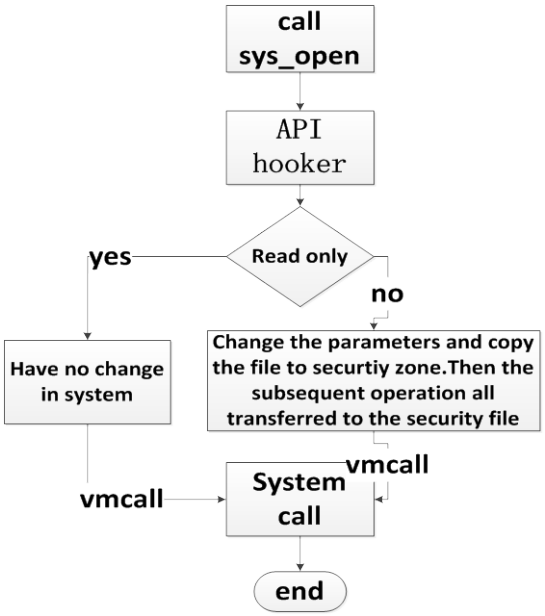

Fig.3. Monitoring of System Call Open

Process isolation: application program can open new process through sys_execve function. Malicious programs can invade into the system through the call. Therefore it also belongs to monitored demand. The monitoring diagram of system call execve in sandbox is shown as follows:

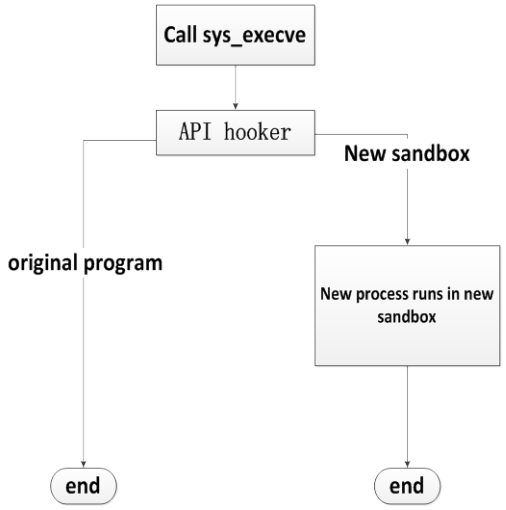

Fig.4 Monitoring of System Call Execve

File rollback: We intercept sys_exit during normal exit of sandbox and check whether the system file is changed by the program or not. The files in the safety area can be copied and duplicated back according to the saved path.

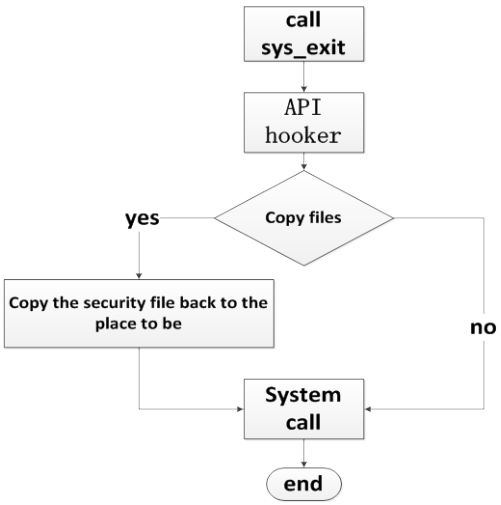

Fig.4. Monitoring of System Call exit

C. Test

1) Functional Test

File redirection test:

Program test_open_app is operated in sandbox. It can open file openfile on the desktop, and text hello can be input inside. Obviously, the operations are intercepted in the sandbox, and writing operation is positioned into file of safety file folder, and the original file is still not changed.

The intercepted operation is restored during safe exit of sandbox, and then 'hello' appears.

2) Performance Test

The conversion key from VMX non-root mode to root mode was available here, therefore the overhead was tested.

We compared performance of VMware Player and sand box operated in dune module.

Performance comparison among lighted operated in dune sandbox, VMware player and that under Linus is shown in the following figure.

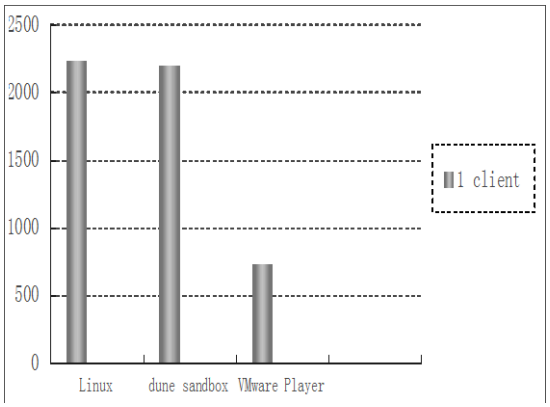

Fig.5. Performance Comparison

Ordinate in the above drawing represents number of response requests per second. It is obvious that program operated under sandbox is nearly the same as the performance operated under Linux, which is nearly three times of VMware Player.

We compared performance losses of operating SPEC2000 between sandbox based on user-level virtualization and NaCI (Native Client) sandbox in the following table.

TABLE II. SPEC2000 TEST

SPEC2000	Sandbox based on dune	NaCI
vpr	4.2%	4.3%
mcf	20.9%	0.8%
ammp	10.1%	1.5%
parser	0.8%	1.6%
perlbmk	1.25%	13%
gap	1.8%	2.4%
vortex	4.4%	11%
Bzip2	1.6%	13%
others	Under 0.5%	

The above table shows that performance loss of program operated in sandbox is lower with average loss performance of 2.9%. However, there are also larger performance losses, such as mcf and ammp since EPT overhead can be caused due to higher TLB failure.

NaCI performance loss in mcfj,ammp is lower than that of dune sandbox prominently. Its average loss performance is

5.95%, and the loss performance is almost 2 times of dune sandbox.

V. CONCLUSION

Sandbox is applied in detecting malicious code more and more maturely as information security technology of efficient isolation. Therefore we created a sandbox on the user-level virtualization platform, therefore it is more smaller. Sandbox can be operated under privilege mode of VMX non-root mode through loading Dune module, switching frequency from application layer to kernel layer in the monitoring process was avoided, thereby keeping higher performance. It has the disadvantage that sandbox function should be further increased and improved.

Acknowledgment

Thanks our teacher for guiding work of the paper.

References

[1] Belay A, Bittau A, Mashtizadeh A. Dune: Safe User-level Access to Privileged CPU Features[J]. Osdi, 2012.(02):61-63

[2] Zhu Hongwei. Study of virtualization security key technology [D]. Master's Thesis. Zhejiang University. 2008

[3] Shi Guang. Study of processor virtualization technology based on VT - x [D]. Master's Thesis. PLA Information Engineering University, 2010

[4] Liu Ya, Li Guangxin, Zhou Lihua. Study of key technologies for API function interception [J]. Microcomputer Development. 2004. (08): 58-60

[5] Zhang Haipeng. Analysis of malicious code behavior [D]. Master's Thesis, Nanjing University of Posts and Telecommunications. 2013

[6] Xie Jiangyan. A sandbox mechanism based on lightweight virtualization [D]. Master's Thesis, Hunan University. 2012

[7] Lin Qiaomin. Research and practice of virtual machine related technology research and practice [D]. Master's thesis. Hohai University. 2004

[8] Li Wei. Study of mechanism of the virtual machine [D]. Master's Thesis, University of Electronic Science and Technology. 2004

[9] Du Hai, Process monitoring method based on full virtualization [J]. Computer Engineering. 2009. (8):88-90

[10] Cao Xin. Analysis and study of paravirtualization technology [D]. Master's Thesis. Zhejiang University. 2008