International Journal of Grid and Distributed Computing
Vol. 9, No. 6 (2016), pp.127-136
http://dx.doi.org/10.14257/ijgdc.2016.9.6.13

Multi-layered Process Isolation Management in Virtual Environment

Tian Junfeng[1], Zheng Junfang[2],Cai Hongyun[3]

[1, 2, 3] *College of Computer Science and Technology, Hebei University, Baoding, China, 071002*
[1]*tjf@hbu.edu.cn*,[2]*zhengjf_13@163.com*, [3]*chy_hbu@126.com*

Abstract

For the problem that the process running is not security in the virtual environment, a multi-level process isolation management scheme is proposed. In this scheme, the processes and the basic runtime environment is defined as the process space. It takes a level hierarchical division strategy to classify the processes from the process space into two levels. One of which is the level of credible processes and another is the level data confidential processes. Under the strategy, the processes operate with different permission to ensure the security of sensitive information, prevent data leaking, operate legally and operate under authority. Finally simulation experiment results show that the method has a good protection to the process running in a virtual environment and it can meet the needs of data confidentiality. At the same time, the overhead of running processes is not obvious.

Keywords*: virtual environment; multi-level; process isolation; credible level*

1. Introduction

Currently, Cloud computing is one of the hottest topics in the area of information technology. Virtualization is one of the main support of cloud computing technology, and its security level is closely related to cloud computing security. With the development and application of virtualization in the cloud computing environment, it allows a platform to run multiple operating systems. At the same time, it brings challenges to ensure the data security. The isolation and the confidentiality of consumer processes are under threat. How to ensure the system operating under data safety becomes a problem that shall be solved immediately.

2. Related Work

Focused on the issue the virtual environment are many security problems, experts and scholars give a lot of solution. Literature[1-2] put forward a novel network framework for cloud computing in the virtual environment of virtual network security to control the virtual machine deployment in the internal communication between the physical machine and higher security. Literature [3] evaluated the security threat for the cloud system under the virtual environment with multiple users. Some scholars took the strategy of key management and encryption to protect the processes under the cloud environment[4-6]. At the same time some scholars focused on the safety of the virtual machine live migration putting forward some major security and trust in the cloud environment adopting a security mechanism[7-8].

At present, fine-grained security is detailed at aspects of procedures. Literature [9-11] brought up the idea that procedures could be monitored to prevent process

ISSN: 2005-4262 IJGDC

passing the analysis maliciously. Researchers also tried the original system to increase security component or building security environment to strengthen the protection ability of the process [12-13].

While the above schemes can effectively ensure the security of a cloud computing process under virtual environment, literature [13] didn't further subdivided each security domain, although literature [12] added sensitive information to protect, there is no hierarchical sensitivity information protection has limitations, fastenings which caused the normal exchange information block. Literature [14] introduced an access control system to ensure the security of the system data.

This paper designs a scheme HPIS(Hierarchical Process Isolation Scheme). HPIS dividing processes into levels according to the credible of the central subject and the data confidentiality. Under the scheme, authority of the process will be subdivided to ensure the security of sensitive information, prevent data leaking, operate legally and operate under authority.

3. Hierarchical Process Isolation Scheme(HPIS)

3.1. Related Definition

Definition 1 PS(Process Space) refers to the safe space environment constructed by processes and the basic runtime environment of processes. The basic runtime environment of processes consists of code segment, data segment.

Definition 2 DCM (Data Control Module) is the core component of this model. This component is the PS management module added in VMM. Then DCM management PS all the way, to ensure the security of the PS and the security of interaction between the PS.

Definition 3 PSM (Process Spatial Message).Supposing $PSM_j = \{ID_j,\ TL_j,\ n_j, CL_j\}$. It is the external characteristics of the PS describing the changes of the processes when PS *j* is running. DCM module uses PSM to control and manage the PS. The structure of PSM diagram is shown in Figure 1.

ID	TL	n	CL(0)	CL(1)	CL(2)	CL(3)

Figure 1. Process Spatial Message

Each process has a unique ID, which uniquely represents a PS. TL(Trusted Level) represents the trust level of a PS. During the initialization of the PS, the system will be in accordance with the establishing rules for DCM allocating confidence levels. *n* represents the corresponding credibility value of a process. The value is used to determine the process level dynamically according to the behavior of a running process. CL(Confidentiality Level) is an abbreviation of Level Confidentiality which represents the relevant parameters of the data at different confidentiality levels in the PS. The confidentiality levels are allocated at the initiation of the PS.

The following are descriptions of TL and CL briefly:

1, TL represents the PS trust level. On the basis of literature [15], this paper will divide credible cases into four grades. Set $TL = (Le(x))$, $x \in G$, $G = \{0,1,2,3\}$. The value of x ranges from 0 to 3 meaning that credible rating of the PS is absolutely credible, credible , critical credible or not credible .

2, CL represents the confidentiality level of the process data in the PS. Set $CL = \{Sl(y)\}$, $y \in G$, $G = \{0,1,2,3\}$. These formulae represent the relevant parameters of the data at different levels in the PS, where *Sl(0)* is relevant parameters to the top-secret level data, *Sl(1)* is confidential data of relevant parameters, *Sl(2)* is the secret data of related parameters and related parameters *Sl(3)* as the common data.

When the process is running, the credibility value of the child processes inherit the credibility of their parent, and the child PS data`s confidentiality level are also decided by the characteristics of its own. It is important to note that process with credible level (3) can't create the child process.

3.2. Basic Thought

The overall structure of the HPIS as shown in Figure 2, each virtual domain has multiple process space. The establishment and exchange of information between the PS need to be controlled and managed by DCM module. DCM module analyzes the process behavior and makes the corresponding decision and feedback according to the PSM and the rule set. In order to avoid DCM as a bottleneck, each virtual machine can be regarded as an independent domain, which corresponds to a DCM module.

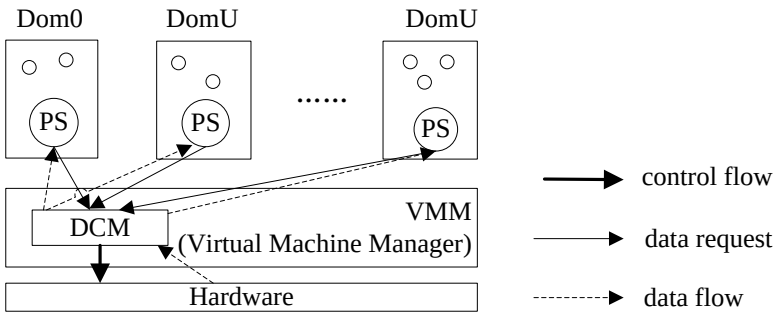

Figure 2. HPIS Integral Structure Diagram

3.3. DCM Module

The logic of the DCM module is shown in the Figure 3.

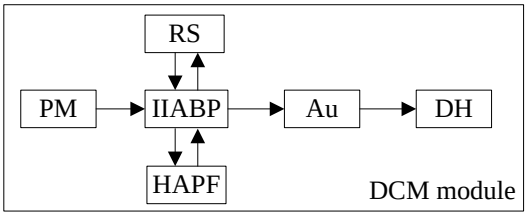

Figure 3. DCM Module's Logical Structure

PM(Process Monitor),is responsible for monitoring the processes' action or behaviors.

IIABP(Information Interaction Analysis Between the Processes) is responsible for judging the legitimacy of the process operation according to the rules in the RS and record of PSM while exchanging information between processes. If the operation is recognized as legitimate, then the process of instruction will be sent to the Au part into the next phase of operation. If the operation is recognized as illegal, then the related information will be sent to the HAPF part to evaluate the confidence level of the process, and the IIABP will refuse the process operation. After waiting for review, the process of a new confidence level results will write back to PSM.

RS(Rule Set) stores the information exchanging rules between processes.

HAPF(Hierarchy Analysis Process Feasible) is responsible for dynamic assessment and adjustment the PS' confidence level according to the evaluation rules .

Au (Authorization),After receiving the analysis result from the IIABP, the Au grants corresponding authority to processes and sends the related instruction parameters to DH.

DH (Data Handling) is responsible for reading data according to the authority granted by Au, then the data is read back to the corresponding PS.

3.4. Information Interaction Rules between Processes

Defined set of operations $Action=\{R,\ W,\ C\}$, wherein R, W and C are Read, Write and Call. And prohibit reading or writing with $\overline{R}$ or $\overline{W}$ said..

For the convenience to discuss, suppose that there is an arbitrary process i as an operating subject and the formula is the PS information. Among them, $TL_i=Le_i(x)$, $CL_i=\{Sl_i(y)\}$, $x\in G$, $y\in G$, $G=\{0,1,2,3\}$. There is an arbitrary process j as operating object, and process j's PS information is represented as $PDM_j=\{ID_j,\ TL_j,\ \eta_j,\ CL_j\}$. Among them, $TL_j=Le_j(x)$, $CL_j=\{Sl_j(y)\}$, $x\in G$, $y\in G$, $G=\{0,1,2,3\}$.

In order to facilitate understanding, the following will explain the symbols:

a. x_i means the feasible level of the process i is x.

b. y_{ij} have double subscript, which means that the process j's data is read by the process i. The confidentiality level in the process i.

c. Suppose there is a variable T, T represents the difference between the trusted level of the process operation subject and the object level. For example, in the formula $T_{ij}=x_i-x_j$, T_{ij} is expressed as a trusted level difference between process i and process j. $T_{ij}>0$ indicates that the process i is less reliable than the process j.

Rule1 For the object process j, if $y_j=0$, then $PD_i \xrightarrow{\overline{R}} PD_j$.

For the top-secret level data of object process j cannot be read by process i.

Rule2 For the object process j, if $y_j>0\bigcup T_{ij}\le 0$, then $PD_i \xrightarrow{R} PD_j \bigcup y_{ij}=y_j$.

When the object process j's data is not the top-secret and the process i's value is greater than or equal to the process j's credibility, process i can read process j's data, and the data in the process i confidentiality level remains the same.

Rule3 For object process j, if $y_j>0\bigcup T_{ij}>0$, then $y_{ji}=y_j-T_{ij}$, at this time if $y_{ji}>0$, then $PD_i \xrightarrow{R} PD_j \bigcup y_{ij}=y_{ji}$ otherwise $PD_i \xrightarrow{\overline{R}} PD_j$.

When the object process j's data is not the top-secret, and the process i is less reliable than the process j, HPIS need to increase the confidentiality level of data. If process i want to read what belongs to the process j, The rule determine whether to allow the process i read. After that, there will be two kinds of cases. First, if after increase the data confidentiality level is top-secret, then process i denies access to the data. The second, if after the upgraded level of confidentiality is not top-secret, allow the process i to read it, and its confidentiality grade in the process i is the confidentiality level upgraded.

Rule4 if $x_i=3$, $PD_i \xrightarrow{\overline{R}} PD_j$.

If the process is not credible, other processes will reject any of its read requests directly.

Rule5 For $\forall PD_i,\forall PD_j$, there are $PD_i \xrightarrow{\overline{W}} PD_j$

Any process does not allow others to write operations. This is to ensure the safety of PS data field from being tampered by other processes.

Rule6 If $PD_i \xrightarrow{C} PD_j$, $PD_j \xrightarrow{rules}$ PDi which rules on behalf of the above Rule 1 to Rule 5.

Process *j* is invoked by process *i*, and can read process *i*'s data according to the rules 1 to 5. If process *i* need to write operation, the results to is transfer by process *j*, and written by itself.

3.5. Confidence Level of Process

HPIS is based on processes' behaviors to measure the confidence level of a process dynamically. According to the data exchanging rules between processes, CL of process is going to be implemented dynamically. If there is any violation of the rules of behavior, the process will be deemed illegal, HPIS need to stop the process and re-evaluate the trust level of the process. The processes' dynamic confidence level judgment diagram is shown in Figure 4.

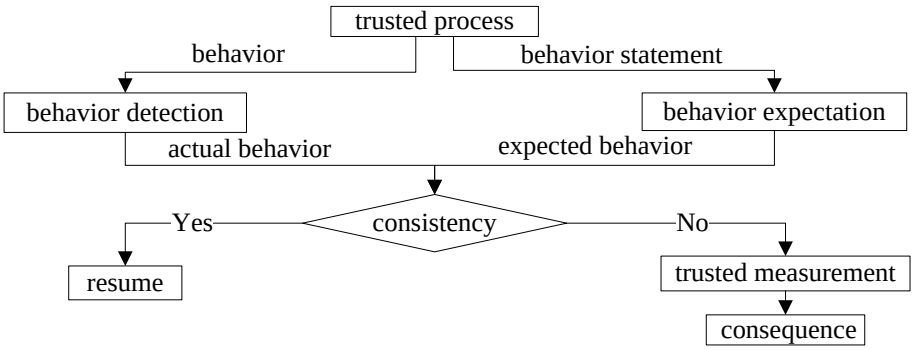

Figure 4. Dynamic Credible Level Judging Process Schematic

Setting three key threshold for the confidence level of a process. Suppose that there are thresholds a, b, c, ($0 < c < b < a < 1$). And assuming that the initial credibility value of the process *i* is n_i. For security and fair, the credibility of the initial process is defined as the critical credible, namely $n_i = b$. During the running of the process, if the process *i* need to create a child process *j*, then the credibility of the initial process *j*'s value is the current of process *i*'s reliability value, namely $n_j = n_i$. Process' level of trusted specific classification is as follows:

1. If $n_i < c$, the process *i* is not credible.
2. If $c \le n_i < b$, the process *i* is the critical credibility.
3. If $b \le n_i < a$, the process *i* is credible.
4. If $a \le n_i$, the process *i* is absolutely credible.

If there are illegal operations, various illegal operations to impact the credibility value, the influence of different permission values are given w_1, w_2. Respectively, when a process is going to read other processes. it is not allowed to access any field; w_1 credibility to reduce value. When a process is going to write the other processes, w_2 credibility to reduce value.

For example, assuming that the initial value of the process *i* is n_i. If there are *m* times operations, with x_{i1} times illegal read operations and x_{i2} times illegal write operations , the credibility value of the process *i* is n_i' at this time:

$$n_i' = n_i - (x_{i1} * w_1 + x_{i2} * w_2) \frac{1}{m}$$

According to the above formula, the dynamic variable of the process reliable level can be realized.

4. Simulation Experiment Results Analysis

4.1. Integrated Simulation Experiment Environment

In order to verify the effect of the scheme for the process isolation management, analyze the security of the scheme and impact on the efficiency of the platform. Simulation environment is based on dell PowerEdge R820 server with Xeon processor E5-4603, 32 GB of memory, under the Xen system of 6.5.0 version.

In order to simplify the experiment, the PS of HPIS is defined as the application of the virtual machine and the running environment of the application program. The 3 key thresholds of HPIS evaluation process confidence level are designated as 0.5, 0.65, 0.9, respectively. That is, when $n_i < 0.5$, the process i is not credible; when $0.5 \le n_i < 0.65$, the process i is the critical credibility; when $0.65 \le n_i < 0.9$, the process i is credible; when $0.9 \le n_i$, the process i is credible.

4.2. Safety Analysis

This experiment is based on the Xen security framework XSM. XSM is a reference to the Linux operating system security framework LSM (Linux Security Module) to achieve a universal access control framework. XSM can support a variety of the expansion of the access control model, and do not need to make changes to the rest of the kernel. This scheme select Flask architecture for the experiment of MLS model. Flask is an excellent security architecture implements in which all the hook functions defined in XSM. It can be more flexible to meet the needs of a variety of security policy.

For the anti-attack capability and data confidentiality aspects: the traditional attack methods, such as rootkit attacks, Trojans or other types of privilege escalation attacks, its purpose is to obtain advanced permission level, breaking the operating system security restrictions, thereby illegally obtained or damage security process status. In order to simplify the experiment, experiment select rootkit, and the two virtual machines in two the same user application for comparison. One is the normal user application a1, the other is a malicious user application a2 attacks by rootkit. Initially, both have the same confidence value, $n_{a1} = n_{a2} = 0.6$. The operation sketch map and the variation of the reliability value of the two are shown in the following figure:

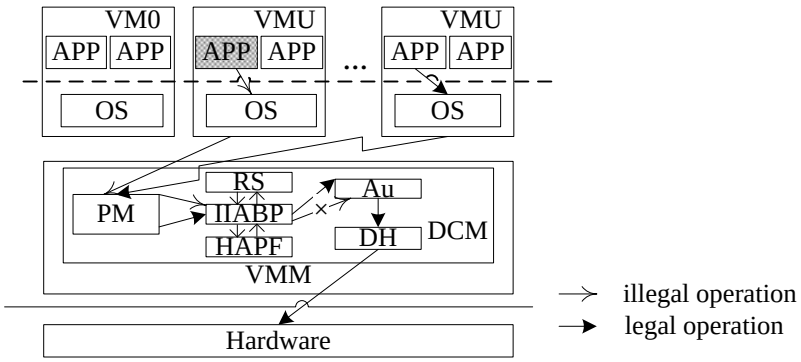

Figure 5. Operation Schematic Diagram

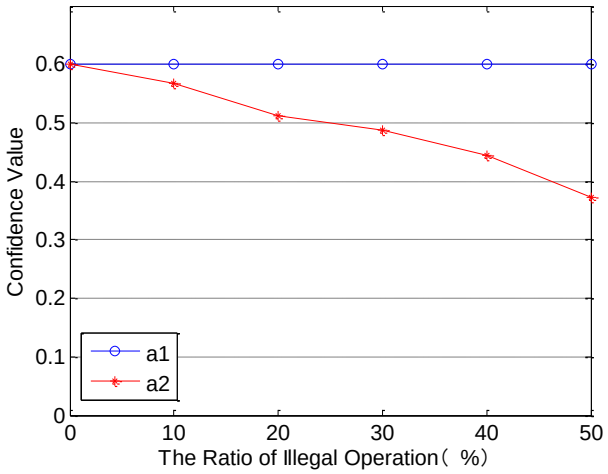

Figure 6. With the Number of Illegal Operation Ratio on the Modifications of the Credibility

There are two applications in Figure 5, one is normal, and the shadow part is infected malicious applications. The legitimate operation of the normal user application according to the arrow refers to the smooth completion of the process. Infected application attempts to modify the operating system code to hide their malicious actions. However, after the PM intercept malicious actions, referred IIABP partially processed and terminates the operation. It can be known that rootkit cannot play a role in the HPIS system by modifying the system kernel data structure or the operation of the instruction execution process.

Figure 6 shows the confidence value change in HIPS, normal user applications and infected malicious user applications. Therefore, a malicious user cannot override read or modify data, and the credibility of the value will fall. Read permissions owned by a malicious user application will be lower. Better data confidentiality are protected.

Firstly, HPIS adopts multilevel security management. The subject with low security cannot read and write the confidential information of the object with high security level, which can effectively prevent information leaking to ensure the confidentiality of object data. Secondly, if the data is allowed to read by the low level security subject, the confidentiality level of the object will be improved. This mechanism can effectively prevent secondary transmission of the data from subject without credibility. Data's flow is more reliable. Finally, HPIS does not allow any subject to write on the object's information, which will effectively restrain and prevent data being tamperied from the malicious process. So, the tampering attack cannot effectively attack the object user. Traditional way of attacking, such as a rootkit attacking, Trojan attacks, and other types of elevated privileges, which purpose is to obtain high level, break through the operating system's security restrictions, and thus illegal to steal or destroy the security process status. But these attacks in process isolation management scheme are losing the ability to attack.

4.3. Performance Analysis

Execution cost is a very important evaluation parameters for systems.If the implementation of the program is too large, the system and process operation will be severely disturbed. In order to test the performance overhead of HPIS, this paper has designed an experiment in two kinds of environments which are the prototype system

(Xen) and the HPIS system (HPIS-Xen). The following tests were carried out in three areas:

Test 1: in Xen and HPIS-Xen . The execution cost of the system main function, such as reading function, system call and hyper call. The results of test 1 are shown in Table1 :

Table 1. Function Call Overhead (ms)

	read function	system call	hyper call
Xen	0.308	0.2	0.273
HPIS-Xen	0.308	0.21	0.302

Test 2: in Xen and HPIS-Xen . The system is mainly the I/O bandwidth, such as system of reading/writing bandwidth, network bandwidth and pipe bandwidth.The results of test 2 are shown in Table 2:

Table 2. System Main I/O Bandwidth (MB/s)

	read/write bandwidth	network bandwidth	pipe bandwidth
Xen	38.7	1814.3	2450.6
HPIS-Xen	31.5	1730.6	2390.3

Test 3: in Xen and HPIS-Xen. The module loading time, such as Xen module, VM1, VM2.The results of test 3 are shown in Table 3 :

Table 3. System Main Module Loading Time (ms)

	Xen Module	VM1	VM2
Xen	2115	721	854
HPIS-Xen	2511	852	974

The above table shows the results of the function execution overhead and I/O bandwidth and module load time in three tests are expressed above tables. Each part of the data is recorded by the table. According to the data, HIPS has little effect on system overhead. HIPS does not affect the system efficiency and user experience.

5. Conclusion

Based on virtualization technology is used in a cloud environment, one of the many security problems of processes safety issues are analyzed. This paper puts forward the solutions HPIS to solve the process runtime isolation and its data confidentiality issues. The method of process running in a virtual environment with good protection by simulation experiments shows that the operational impacts performance overhead of the process are not obvious. Next, the method will be further optimized. The scheme will be further optimized and improved to operating better as well.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61379116), the Natural Science Foundation of Hebei province (F2016201244,F2014201165) and the Natural Science Foundation of Hebei institution (ZD2015088).

References

[1] H. Wu, Y. Ding, C. Winer and L. Yao, Network security for virtual machine in cloud computing. In Computer Sciences and Convergence Information Technology (ICCIT), International Conference on. IEEE.(2010)
[2] S. S. Pandya, Security for virtual machine in cloud computing[J]. International Journal of Management, IT and Engineering.(2013)
[3] A. Jasti, P. Shah, R. Nagaraj and R. Pendse, Security in multi-tenancy cloud. In Security Technology (ICCST), International Carnahan Conference on. IEEE. (2010)
[4] R. Chandramouli, M. Iorga and S. Chokhani, Cryptographic key management issues and challenges in cloud services. Springer New York.(2014)
[5] O. D. Alowolodu, B. K. Alese, A. O. Adetunmbi, O. S. Adewale, and O. S. Ogundele, Elliptic curve cryptography for securing cloud computing applications[J]. International Journal of Computer Applications.(2013)
[6] G. Kappes, A. Hatzieleftheriou and S. V. Anastasiadis. Virtualization-aware access control for multitenant filesystems. In Mass Storage Systems and Technologies (MSST), Symposium on. IEEE. (2014)
[7] M. R. Anala, J. Shetty G. and Shobha , A framework for secure live migration of virtual machines. In Advances in Computing, Communications and Informatics (ICACCI), International Conference on. IEEE. (2013)
[8] M. Aslam, C. Gehrmann and M. Bjorkman, Security and trust preserving vm migrations in public clouds. In Trust, Security and Privacy in Computing and Communications (TrustCom), International Conference on. IEEE. (2012)
[9] F. Baiardi, D. Maggiari, D. Sgandurra and F. Tamberi, Transparent process monitoring in a virtual environment[J]. Electronic Notes in Theoretical Computer Science.(2009)
[10] X. Liu, L. Cao, M. Liu and K. Han, Process-based Security Detection Approach for Virtual Machines on Private Cloud Platforms[J]. Journal of Networks.(2013)
[11] H. DU and R. CHEN,Full-virtualization-based Process Monitoring Method[J].Computer Engineering.(2009)
[12] X.Q. LI, X.D. ZHAO and Q.K. ZENG, One-Way Isolation Execution Model Based on Hardware Virtualization[J].Journal of Software.(2012)
[13] B. ZHAO, Z.L. XIA, Y. AN and S. XIANG, Research and implementation of process isolation under virtualization environment[J]. Huazhong Univ.of Sci.& Tech.(Natural Science Edition).(2014)
[14] G. Zhu and H. Huang, RESEARCH AND DESIGN OF A MULTILEVEL INFORMATION ACCESS CONTROL SYSTEM[J]. Jisuanji Yingyong yu Ruanjian.(2013)
[15] J.F. TIAN and Y.J. ZHANG, Checkpoint trust evaluation method based on Markov[J]. Journal on Communications.(2015) 36(1)

Authors

Tian Junfeng. (1965-) Male. PhD, professor, and PhD supervisor. His current research interests distributed computing, network technology, trusted computing, cloud computing, *etc*.

Zheng Junfang. (1989-) Female. Master. Her current research interests include cloud computing, trusted computing, *etc*.

Cai Hongyun. (1980-) Female. Master and assistant professor. Her main research interests include trust computing and recommender systems.