International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:9, 2012

# A Software of Intrusion Detection Mechanism for Virtual Platforms

Ying-Chuan Chen, Shuen-Tai Wang

***Abstract*****—**Security is an interesting and significance issue for popular virtual platforms, such as virtualization cluster and cloud platforms. Virtualization is the powerful technology for cloud computing services, there are a lot of benefits by using virtual machine tools which be called hypervisors, such as it can quickly deploy all kinds of virtual Operating Systems in single platform, able to control all virtual system resources effectively, cost down for system platform deployment, ability of customization, high elasticity and high reliability. However, some important security problems need to take care and resolved in virtual platforms that include terrible viruses, evil programs, illegal operations and intrusion behavior. In this paper, we present useful Intrusion Detection Mechanism (IDM) software that not only can auto to analyze all system's operations with the accounting journal database, but also is able to monitor the system's state for virtual platforms.

***Keywords*****—**security, cluster, cloud, virtualization, virtual machine, virus, intrusion detection

## I. Introduction

CLOUD computing [1, 2, 3] is a novel model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (including servers, storage, networks, applications, data, software, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. It is the access to computers and their functionality through difference deployment models like public, private, and hybrid. Users of the cloud request this access from a set of web service interfaces that manage and monitor a pool of computing resources base on the virtualization technology including virtual machines, network devices, storage, operating system, application programs and development environments. When granted service requests from users, a part of the computing resources in the resource pool is dedicated to the requesting user until those resources are released.

Moreover, users can not actually see or specify the physical location and organization of the equipment hosting the resources, they are ultimately allowed to use base on any types of the cloud service models in figure 1, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). There are many benefits with cloud technology including low cost, simple management, high scalability, high reliability, facilitation, on-demand, pay-per-use, and customization.

Y.C. Chen is with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: ycc0301@nchc.narl.org.tw).
S.T. Wang is with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: stwang@nchc.narl.org.tw).

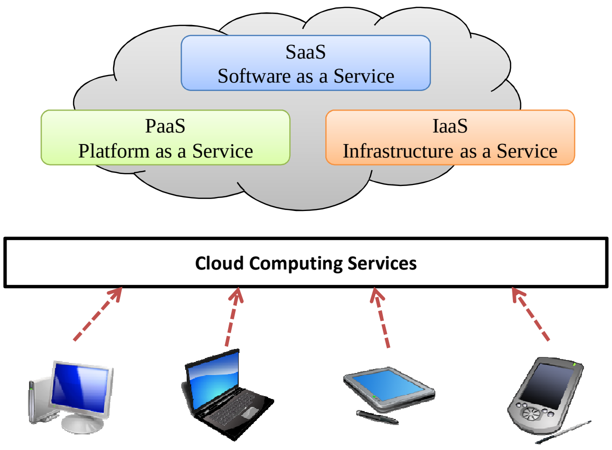


Fig. 1 Cloud service models

- IaaS - it is a provision model in which an organization outsources the equipment used to support associated operations, such as storage, hardware, servers and network components. This service model allows the cloud to operate during high traffic and demanding situations as resources are dynamically increased as they are needed. Typical examples are Amazon EC2 (Elastic Cloud Computing) Service [4] and S3 (Simple Storage Service) [5] where compute and storage infrastructures are open to public access with a utility pricing model.
- PaaS - it is the delivery of a virtualized application, computing platform and solution stack as a service. It offers a high-level integrated environment to build, test, and deploy custom applications. All applications and infrastructure are run and managed by the services vendor. Generally, developers will need to accept some restrictions on the type of software they can write in exchange for built-in application scalability. Google's App Engine [6] is a well-known service for PaaS style, which enables users to build Web applications on the same scalable systems that power Google applications.
- SaaS - it is a software delivery method that provides access to software and its functions remotely as a Web-based service. Providers offer users access to specific application programs controlled and executed on the provider's infrastructure, such as Salesforce [7] is an industry leader in providing online CRM (Customer Relationship Management) Services. And Windows Live Mesh [8] by Microsoft Company allows files and folders to be shared and synchronized across multiple devices from anywhere.

Furthermore, there are three fundamental categories of cloud deployment models including public, private and hybrid cloud.

- Public cloud - the cloud mode is open for use to the general public, whether individuals, corporations or other types of organizations. Where multiple customers can access web applications and services over the internet. Each individual customer has their own resources which are dynamically provisioned by the cloud service provider that hosts the cloud for multiple customers from multiple data centers, manages all the service security problems and provides the infrastructure for the cloud to operate stably. The customer has no control or insight into how the cloud is managed or what infrastructures are available. This deployment model is the most common cloud type in market.
- Private cloud - the cloud environment within the boundaries of an organization and typically for its exclusive usage. It follows the concept of cloud computing on a private and internal network. A lot of business companies or organizations adopt this model to build own internal could platform. It lets customers to have the benefits of cloud computing without some of the pitfalls, and grants complete control over how data is managed and what security measures are in place. This can offer to users having more confidence and control. Besides, the major issue is that the users have large expenditures as they have to buy related infrastructure to run the cloud platform and also have to manage the platform themselves using this deployment model.
- Hybrid cloud - this is a composition of two cloud mode including private and public that remain unique entities but are bound together, offering the benefits of multiple deployment models. It allows users to hold sensitive information on private model, and handle large traffic and demanding situations with the public model. Hence, it can benefit from both deployment models for users.

The Virtualization Technology (VT) acts a very important role that can achieve the purpose of cloud platforms and services, and it is the ability to run many virtual machines on top of a hypervisor. A virtual machine (VM) [9, 10, 11, 12] is a software implementation of a machine that executes related programs like a physical machine. Each VM includes its own system kernel, OS, supporting libraries and applications. A hypervisor provides a uniform abstraction of the underlying physical machine, and multiple VMs can execute simultaneously on a single hypervisor. The decoupling of VM from the underlying physical hardware is able to allow the same VM to be started and ran on different physical environments. Thus virtualization is seen as an enabler for cloud computing, allowing the cloud service provider the necessary flexibility to move and allocate related computing resources requested by the user wherever the physical resources are available.

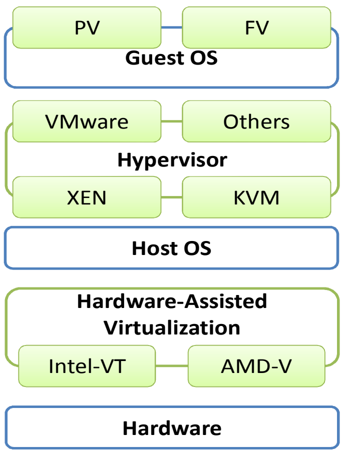


Fig. 2 Virtualization architecture

In figure 2, it shows the principal concept and architecture of virtualization technology. Physical hardware resources were divided as virtual resources of virtualization platforms by VM's monitors, and those virtual hardware resources were assigned to each VM by different requirements. Moreover, The VM's monitor acts an importance role to connect between hardware and virtual devices, providing each Guest OS a set of virtual interfaces. All instructions were delivered to hardware layer from virtual layer, which receives results from hypervisor of VMs. For Guest OS, there are two central virtualization types [13, 14, 15] including para-virtualization (PV) and full-virtualization (FV), which can be both combined with hardware-assisted virtualization.

- With para-virtualization model, Guest OS is simulated by modified Kernel of Linux, but related system's devices are not emulated. Instead, all devices are accessed through a set of light-weight virtual drivers offering better system performance and close to the physical machine. But the drawback is that guest kernels must be upgraded to provide new virtual system calls for the new services and all of Guest OS must be compatible with the Host OS.
- Full-virtualization model allows the execution of unmodified virtual OS by emulating physical system's hardware resources. The advantage of FV is flexibility that it can build various virtual platforms in single physical machine. Nevertheless, FV may incur a performance penalty. The VMM of FV have to offer the VM through an image of entire system, including virtual memory space and devices, that also must create and maintain data structures for virtual components, such as a shadow memory page table, and these data structures should be updated for every corresponding access with the VMs.

Both Intel and AMD corporations supported the hardware-assisted virtualization modules for their novel products of CPU. The Intel module, like the Intel-Virtualization Technology (Intel-VT) [16, 17, 18], combined with software-based virtualization solutions provides maximum system utilization by consolidating multiple environments into a single physical machine.

For AMD module, like AMD Virtualization (AMD-V) [19, 20, 21] Technology, allows users to better utilize related resources, which make virtual platforms more effective.

It is an evolving sub-domain of computer security, network security and information security for cloud computing security [22, 23, 24] that refers to a broad set of policies, technologies and controls deployed to protect data, applications and the related infrastructure of cloud platform. A lot of security issues associated with cloud computing area, which fall into two broad categories include security issues faced by cloud providers which organizations providing IaaS, PaaS or SaaS through the cloud computing technology and security issues faced by cloud customers. In general, the cloud service provider must ensure that their infrastructure or platform is secure and compliant. Then anytime let customers' secret, data and applications are protected.

Base on cloud computing security issues, in order to ensure that virtualization platform is properly secured, we appear an integrated intrusion detection mechanism for illegal intrusion and behavior for virtualization platform.

## II. Intrusion Detection Mechanism

In order to ensure the system is secured and cleared, we have to discover and resist any illegal intrusion or behavior in the virtualization platform. Following this idea, we demonstrate a useful detection mechanism for illegal intrusion that not only analyze the journal for platform's accounting, but also monitor the security status of system. In this section will discuss related concepts for virtualization system's security, and display the specific implement architecture for our methods.

### *A. Intrusion Detection Package*

Intrusion Detection Package (IDP) is a useful detection package that can check file and directory of system where is secure and integrity. The concept of IDP is following that it constructs a new database by clear system from the regular expression rules that can be defined in the configuration file (IDP.conf). Once this database is initialized, and it can be used to verify the integrity of the files or directories. Several well-known message digest algorithms are adopted that not only compare and check the integrity of the file or directory, but also various file attributes can be checked for inconsistencies. The package features [25] following:

- The message digest algorithms including md5, sha1, rmd160, tiger, crc32, sha256, sha512, and whirlpool are supported.
- Related file attributes are supported include file type, permissions, inode, uid, gid, link name, size, block count, number of links, mtime, ctime and atime.
- Posix ACL, SELinux, XAttrs and Extended file system attributes are supported.
- Support plain text configuration files and database for simplicity.
- The powerful regular expression supports selectively include or exclude files and directories to be monitored.
- Support the gzip format for database compression.

Generally, the administrator will construct clear database on a secure system before it is brought onto the network. This clear database is a snapshot of the system in secure state and the yardstick through which all subsequent updates and changes will be analysis. The database should cover significant information like key system binaries, libraries, header files and all files that are expected to remain the same over time. Nevertheless, it should not contain useless information for files which change frequently such as log files, mail spools, proc filesystems, home or temporary directories.

After a break-in, administrator may begin by examining the status of system using associated tools like ls, ps, netstat and who, those commands of system most likely to be trojaned. For example, imagine that ls has been doctored to not display any illegal log files, and that ps command have been rewritten to not display any information for any illegal processes. Even the administrator who had previously printed out on paper the dates and sizes of these key system files cannot be certain by comparison that they have not been modified in some way. All of file dates and sizes can be manipulated; a hacker can make this easy and trivial by some powerful rootkit tools. Therefore, it is possible to manipulate file dates and sizes, but it is much more difficult to manipulate a single cryptographic checksum like md5, and exponentially more difficult to manipulate each of the entire arrays of checksums. The administrator can quickly by rerunning IDP identify changes to important files or directories.

### *B. Processing Accounting Package*

Processing Accounting Package (PAP) is a useful package for monitoring users and applications, the package works in the background of system recording what users are doing on the system as well as the associated system resources that are being consumed. Hence, it can provide an excellent solution which told administrator not only how long the user was on the system but what commands user was using and how much of the system resources were using. It contains several great utilities for monitoring process activities, including ac, lastcomm, accton and sa.

- The ac [26] command shows statistics about how long users have been logged on the system.
- The accton [27] command is used to turn on or off process accounting.
- The lastcomm [28] command can shows related information about previously executed commands.
- The sa [29] command is used to display related summarizes accounting information about previously executed commmands.

Process accounting support with PAP package has been integrated into the newer kernels of Linux OS, which kernel version greater than or equal to version 1.3.73 is required.

### *C. Architecture*

Figure 3 demonstrations the main Architecture of our intrusion detection mechanism; we integrate the firewall, IDP, PAP and related system's tool to our intrusion detection mechanism for virtualization platforms.

This mechanism can discover any illegal intrusion then send the report to system's administrator by e-mail mode that analyzes user's or hacker's illegal behaviors in virtualization platforms.

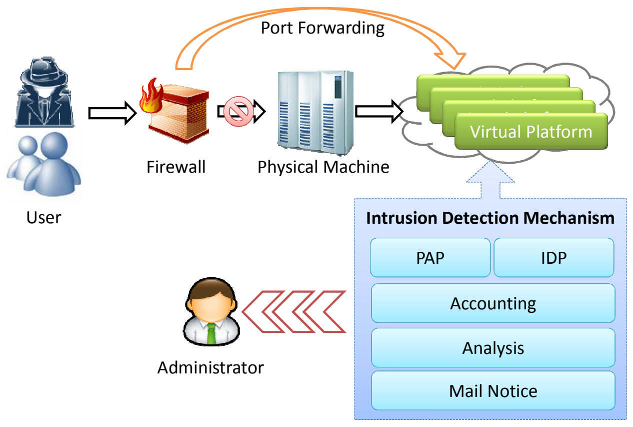


Fig. 3 Architecture of intrusion detection mechanism

In firewall technology, the iptable [30] daemon is a very powerful administration tool for IPv4 packet filtering and Network Address Translation (NAT) [31], the common expression likes following:

```
iptables [-t table] - CMDS chain rule - specification [options]
```

It is used to filter all IP address that will resist all illegal connections for any network interface devices, and restrict any useless service ports of virtualization platforms. Furthermore, we provide specific tunnel that let users connect to their virtual platforms by particular port using port forwarding technology can directly prevent intrusion and destruction in the physical machine, following below command:

```
iptables - A INPUT - p tcp - i eht0 - -dport 9000 - j ACCEPT ;
iptables - t nat - A PREROUTING - p tcp - -dport 9000 \
          - d 140.x.x.x - i eth0 - j DNAT - -to 192.168.0.101 : 22 ;
```

For IDP, it can check the image file of virtual machine whether is clear and secure following below:

- Mount VM image – There is a very important image file (likes filesystem.img) for the virtualization platform need to check. Kpartx [32] is a usefully tool for mounting partitions within an image file, it can be used to set up device mappings for the partitions of any partitioned block devices, then using mount command to mount vm.img to be file and directory type, such as mount -o loop /vm.img /vm. After mounting, we can see related contents in virtual disk including /vm/etc, /vm/bin, /vm/root, /vm/home and /vm/opt.
- Setup IDP daemon – We can setup the IDP daemon in host machine to detect status of the virtualization platform, and modify configure file in /etc/idp.conf to define detection rules what matching our application requirements.
- IDP's database initialization – All important files and directories will be compressed to idp.db.new.gz with the gzip compression format. Principal files include /etc/fstab, /etc/shadow, /etc/passwd and /etc/sudoers.
- IDP check – An idp.log will be created after IDP checking process, and then we can know whether the system is secure from log file via checking operations.
- IDP update – Administrator must update IDP's DB if the virtualization platform was changed legally.
- Auto-check mode – building the auto-checking script with the crontab file anytime you like, it able to detect for any states of virtualization platform including running, pause and power off.
- Report – it will mail to administrator if any illegal operations and intrusions were detected with IDP.

For the PAP mechanism, we modify associated programs including accton, pacct and lastcomm, then apply the mechanisms to record and analyze system's logs following below works:

- Accton – modifying the program to control the pacct daemon.
- Pacct – setting the program in /etc/init.d/psacct to check system's and user's operations.
- Lastcomm – setting the program to retrieve and customize associated system's information.
- Building the PAP log database, and setup database in the host machine.
- Setup and start PAP daemon in every virtual machines of the platform.
- Apply auto-checking mode for anytime you like, such as we can let PAP daemon to scan system when power off, and detect illegal operations for users per hour as well.
- Any operations were recorded in the local database. Besides, PAP will send significance detection reports to administrator of the platform that can immediately notify administrator to solve security problems if any illegal operations and intrusions were detected.

In next session, we will demonstrate associated significance experiments for virtualization platform's with our intrusion detection mechanism.

## III. Experiment

In this section, we will perform our intrusion detection mechanism that not only record and analyze the journal for system's accounting, but also monitor the state for system's security on virtualization platform with Kernel-based Virtual Machine (KVM) [33, 34, 35].

TABLE I
Testbed Information

| | |
|---|---|
| CPU | Quad-Core AMD Opteron(tm) Processor 2356 |
| DISK | Western Digital(WD) 500G SATA 7200RPM |

| | |
|---|---|
| MEMORY | 8 GB |
| Swap | 2 GB |
| NETWORK | 1 Gb Ethernet |
| OS | CentOS 5.5 (2.6.18-194) |
| VM | kvm-83-164 |

The specification information of testing platform is showed in Table 1. We adopt two HP servers as the hardware testbed, with two Quad-Core AMD Opteron(tm) Processor 2356 CPU in each server likes. The SATA 500GB disk of Western Digital (WD) is used to allocated 2 GB size as swap space. There is 8 GB memory and 1 Gb Ethernet network interface card (NIC) in each tested physical machine. A physical platform was built as the real cluster environment with CentOS 5.5, and the virtual platform is deployed through the VM software version is 3.0 for kvm.

The figure 4 shows the architecture of testbed, there is a physical server as the front node using public ip likes 140.x.x.x that offers exclusive connection tunnels through the port forwarding technology for virtual platforms. Virtual platforms include n1v1 and n2v1, which the private ip address is 192.168.0.101 and 192.168.0.102 respectively. Those virtual platforms are built from physical platform of n1 (192.168.0.1) and n2 (192.168.0.2) with full-virtualization technology of KVM. We use the network file system (nfs) technology that can share related databases including IDP, PAP, and VM image from n1 server to all virtual machines. Besides, administrator can immediately receive analysis reports of this intrusion detection mechanism if any unfriendly actions or intrusions are discovered anytime.

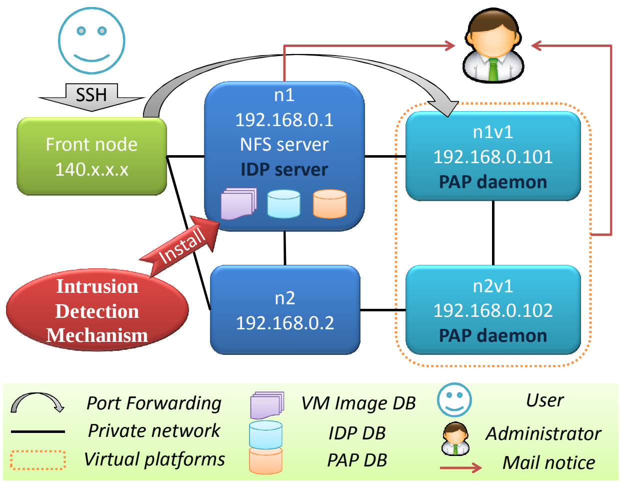
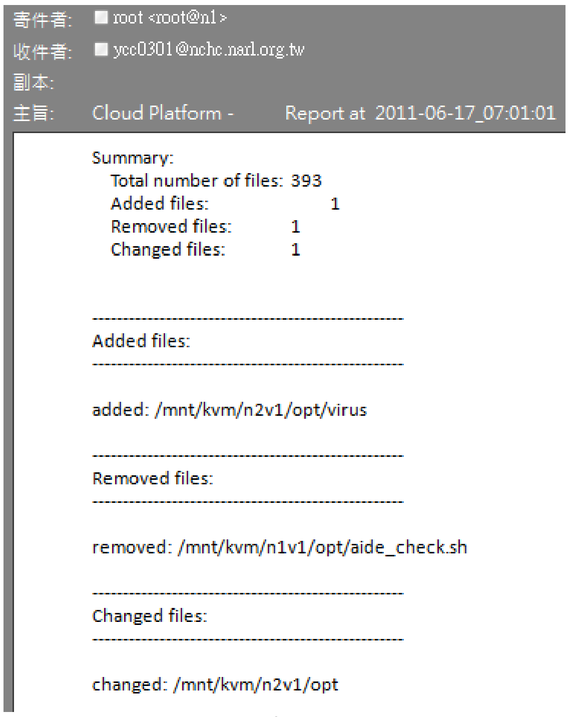


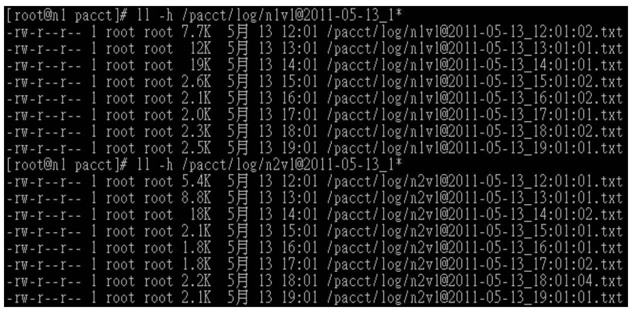
Fig. 4 Architecture for testbed

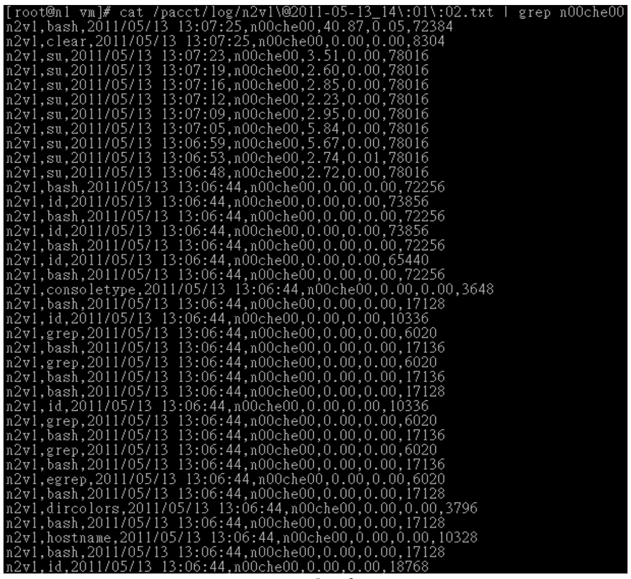

寄件者: root <root@n1>
收件者: ycc0301@nchc.narl.org.tw
副本:
主旨: Cloud Platform - Report at 2011-06-17_07:01:01

```
Summary:
  Total number of files: 393
  Added files:              1
  Removed files:       1
  Changed files:       1


---------------------------------------------------
Added files:
---------------------------------------------------

added: /mnt/kvm/n2v1/opt/virus

---------------------------------------------------
Removed files:
---------------------------------------------------

removed: /mnt/kvm/n1v1/opt/aide_check.sh

---------------------------------------------------
Changed files:
---------------------------------------------------

changed: /mnt/kvm/n2v1/opt
```

Fig. 5 Mail for IDM report

If any illegal status in the virtual testing platform were found, our intrusion detection mechanism can directly analyze illegal status and send a detection report to the administrator of platform. In figure 5, it shows the mail report using mail box (Microsoft Outlook 2010), it display the scan information of platform form system's mail address (root@n1) to administrator's mail address (ycc0301@nchc.narl.org.tw) through the sendmail daemon at 2011-06-17 17:01:01. The mechanism found a virus in the directory (/opt/virus) of n2v1 and discovered an event that a significance program of n1v1 is removed.


```
[root@n1 pacct]# ll -h /pacct/log/n1v1@2011-05-13_1*
-rw-r--r-- 1 root root 7.7K  5月 13 12:01 /pacct/log/n1v1@2011-05-13_12:01:02.txt
-rw-r--r-- 1 root root  12K  5月 13 13:01 /pacct/log/n1v1@2011-05-13_13:01:01.txt
-rw-r--r-- 1 root root  19K  5月 13 14:01 /pacct/log/n1v1@2011-05-13_14:01:01.txt
-rw-r--r-- 1 root root 2.6K  5月 13 15:01 /pacct/log/n1v1@2011-05-13_15:01:02.txt
-rw-r--r-- 1 root root 2.1K  5月 13 16:01 /pacct/log/n1v1@2011-05-13_16:01:02.txt
-rw-r--r-- 1 root root 2.0K  5月 13 17:01 /pacct/log/n1v1@2011-05-13_17:01:01.txt
-rw-r--r-- 1 root root 2.3K  5月 13 18:01 /pacct/log/n1v1@2011-05-13_18:01:02.txt
-rw-r--r-- 1 root root 2.5K  5月 13 19:01 /pacct/log/n1v1@2011-05-13_19:01:01.txt
[root@n1 pacct]# ll -h /pacct/log/n2v1@2011-05-13_1*
-rw-r--r-- 1 root root 5.4K  5月 13 12:01 /pacct/log/n2v1@2011-05-13_12:01:01.txt
-rw-r--r-- 1 root root 8.8K  5月 13 13:01 /pacct/log/n2v1@2011-05-13_13:01:01.txt
-rw-r--r-- 1 root root  18K  5月 13 14:01 /pacct/log/n2v1@2011-05-13_14:01:02.txt
-rw-r--r-- 1 root root 2.1K  5月 13 15:01 /pacct/log/n2v1@2011-05-13_15:01:01.txt
-rw-r--r-- 1 root root 1.8K  5月 13 16:01 /pacct/log/n2v1@2011-05-13_16:01:01.txt
-rw-r--r-- 1 root root 1.8K  5月 13 17:01 /pacct/log/n2v1@2011-05-13_17:01:02.txt
-rw-r--r-- 1 root root 2.2K  5月 13 18:01 /pacct/log/n2v1@2011-05-13_18:01:04.txt
-rw-r--r-- 1 root root 2.1K  5月 13 19:01 /pacct/log/n2v1@2011-05-13_19:01:01.txt
```
Fig. 6 Journal database

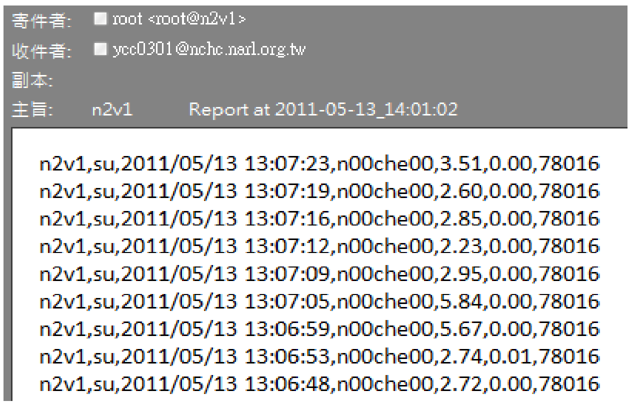


```
[root@n1 vm]# cat /pacct/log/n2v1\@2011-05-13_14\:01\:02.txt | grep n00che00
n2v1,bash,2011/05/13 13:07:25,n00che00,40.87,0.05,72384
n2v1,clear,2011/05/13 13:07:25,n00che00,0.00,0.00,8304
n2v1,su,2011/05/13 13:07:23,n00che00,3.51,0.00,78016
n2v1,su,2011/05/13 13:07:19,n00che00,2.60,0.00,78016
n2v1,su,2011/05/13 13:07:16,n00che00,2.85,0.00,78016
n2v1,su,2011/05/13 13:07:12,n00che00,2.23,0.00,78016
n2v1,su,2011/05/13 13:07:09,n00che00,2.95,0.00,78016
n2v1,su,2011/05/13 13:07:05,n00che00,5.84,0.00,78016
n2v1,su,2011/05/13 13:06:59,n00che00,5.67,0.00,78016
n2v1,su,2011/05/13 13:06:53,n00che00,2.74,0.01,78016
n2v1,su,2011/05/13 13:06:48,n00che00,2.72,0.00,78016
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,72256
n2v1,id,2011/05/13 13:06:44,n00che00,0.00,0.00,73856
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,72256
n2v1,id,2011/05/13 13:06:44,n00che00,0.00,0.00,73856
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,72256
n2v1,id,2011/05/13 13:06:44,n00che00,0.00,0.00,65440
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,72256
n2v1,consoletype,2011/05/13 13:06:44,n00che00,0.00,0.00,3648
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17128
n2v1,id,2011/05/13 13:06:44,n00che00,0.00,0.00,10336
n2v1,grep,2011/05/13 13:06:44,n00che00,0.00,0.00,6020
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17136
n2v1,grep,2011/05/13 13:06:44,n00che00,0.00,0.00,6020
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17136
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17128
n2v1,id,2011/05/13 13:06:44,n00che00,0.00,0.00,10336
n2v1,grep,2011/05/13 13:06:44,n00che00,0.00,0.00,6020
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17136
n2v1,grep,2011/05/13 13:06:44,n00che00,0.00,0.00,6020
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17136
n2v1,egrep,2011/05/13 13:06:44,n00che00,0.00,0.00,6020
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17128
n2v1,dircolors,2011/05/13 13:06:44,n00che00,0.00,0.00,3796
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17128
n2v1,hostname,2011/05/13 13:06:44,n00che00,0.00,0.00,10328
n2v1,bash,2011/05/13 13:06:44,n00che00,0.00,0.00,17128
n2v1,id,2011/05/13 13:06:44,n00che00,0.00,0.00,18768
```

Fig. 7 Journal information

Fig. 6 shows the journal database with our intrusion detection mechanism, it is able to record associated system's operation of all users, and then it will create files to record those logs per hours. The system's administrator can easy retrieve all command journal information of any users for each virtual machine including n1v1 and n2v1 from PAP database that recording logs in physical machine like n1. For example, we can see related journal information of 2011/05/13 for n1v1 and n2v1.

In Fig. 7, it demonstrations related command logs between 13 o'clock and 14 o'clock for specific user in n2v1. The journal structure of file is following hostname, command, date, user, execution time, time of CPU and memory. There is a user who names n00che00, and the user executed commands including clear, su, grep, and hostname at 2011/05/13. We can discover user execute illegal commands and operation from this journal information. Moreover, administrator can apply accounting work for each user by those usefully journal information. For instance, we can calculate how many CPU resources are used for each user.



寄件者: root <root@n2v1>
收件者: ycc0301@nchc.narl.org.tw
副本:
主旨: n2v1 Report at 2011-05-13_14:01:02

```
n2v1,su,2011/05/13 13:07:23,n00che00,3.51,0.00,78016
n2v1,su,2011/05/13 13:07:19,n00che00,2.60,0.00,78016
n2v1,su,2011/05/13 13:07:16,n00che00,2.85,0.00,78016
n2v1,su,2011/05/13 13:07:12,n00che00,2.23,0.00,78016
n2v1,su,2011/05/13 13:07:09,n00che00,2.95,0.00,78016
n2v1,su,2011/05/13 13:07:05,n00che00,5.84,0.00,78016
n2v1,su,2011/05/13 13:06:59,n00che00,5.67,0.00,78016
n2v1,su,2011/05/13 13:06:53,n00che00,2.74,0.01,78016
n2v1,su,2011/05/13 13:06:48,n00che00,2.72,0.00,78016
```

Fig. 8 Mail for PAP report

Fig. 8 shows a PAP report with the mail box, it discover a user who tries to catch permission of root by su command. Our intrusion detection mechanism will automatically send an analysis report to the administrator of platform if any illegal commands are execute by all users. Accordingly, it can immediately help administration to solve all security programs with mail reports, and prevent related illegal operations for virtual platforms.

## IV. Conclusions

In this paper, we present a significant software of intrusion detection mechanism for virtualization systems which likes virtual cluster and cloud platform. The software can not only record the journal for system's operation, but also monitor associated system's security state in virtual platforms. If any illegal proceedings and operations of users are detected anytime, the intrusion detection mechanism will automatically analyze those illegal behaviors, and send a detailed analysis reports mail from the virtual platform to mailbox of administrator who can directly prevent system troubles and solve related security problems. Furthermore, system's administrator can easy to calculate accounting statistics for all users in each virtual machine of platform. Therefore, our intrusion detection mechanism is a very useful and powerful security software for any illegal proceedings or intrusions detection in virtual platforms, and we can obtain interrelated accounting statistics form the database as well.

## References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, April 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," 2009.

[3] I. Foster, Z. Yong, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," *IEEE Grid Computing Environments Workshop*, pp. 1-10, 2008.

[4] Amazon EC2, http://aws.amazon.com/ec2.

[5] Amazon S3, http://aws.amazon.com/s3.

[6] Google App Engine, http://code.google.com/appengine/.

[7] Salesforce, http://www.salesforce.com/.

[8] Live Mesh, http://explore.live.com/windows-live-mesh.

[9] J. P. Buzen and U. O. Gagliardi, "The Evolution of Virtual Machine Architecture," *National Computer Conference Proceedings, AFIPS Press*, vol. 42, pp. 291-299, June 1973.

[10] R. A. Meyer and L. H. Seawright, "A Virtual Machine Time-Sharing System," *IBM Systems Journal*, vol. 9, no. 3, 1970.

[11] R. P. Goldberg, "Architecture of Virtual Machines," *National Computer Conference Proceedings*, *AFIPS Press*, vol. 42, pp. 309-318, June 1973.

[12] R. P. Goldberg, "Survey of Virtual Machine Research," *IEEE Computer*, vol. 7, no. 6, pp. 34-45, June 1974.

[13] L. Nussbaum, F. Anhalt, O. Mornard and J.-P. Gelas, "Linux-based virtualization for HPC clusters," *Linux Symposium*, pp. 221-234, July 2009.

[14] M. Fenn, M. Murphy, and S. Goasguen, "A Study of a KVM-based Cluster for Grid Computing," *47th ACM Southeast Conference*, March 2009.

[15] Wei Chen, Hongyi Lu, Li Shen, Zhiying Wang, Nong Xiao and Dan Chen, "A Novel Hardware Assisted Full Virtualization Technique," *The*

*9th International Conference for Young Computer Scientists*, pp. 1292-1297, Nov. 2008.

[16] Intel-VT, http://www.intel.com/technology/virtualization/.

[17] G. Neiger, A. Santoni et all, "Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization", *Intel Computer Journal*, vol. 10, issue 3, August 2006.

[18] Rich Uhlig, Gil Neiger, Dion Rodgers, Amy L. Santoni, Fernando C.M. Martins, Andrew V. Anderson, Steven M. Bennett, Alain Kagi, Felix H. Leung, Larry Smith, "Intel Virtualization Technology," *IEEE Computer*, vol. 38, no. 5, pp. 48-56, May 2005.

[19] AMD-V, http://sites.amd.com/us/business/it-solutions/virtualization/Pages/virtualization.aspx.

[20] Greg Goth, "Virtualization: Old Technology Offers Huge New Potential," *IEEE Distributed Systems Online*, vol. 8, no. 2, 2007.

[21] Yan Wen, Jinjing Zhao, Huaimin Wang and Jiannong Cao, "Implicit Detection of Hidden Processes with a Feather-Weight Hardware-Assisted Virtual Machine," *ACISP '08 Proceedings of the 13th Australasian conference on Information Security and Privacy*, pp. 361-375, 2008.

[22] B. R. Kandukuri, V. R. Paturi and Atanu Rakshit, "Cloud Security Issues," *2009 IEEE International Conference on Services Computing*, pp. 517 -520, 2009.

[23] Sean Carlin and Kevin Curran, "Cloud Computing Security," *International Journal of Ambient Computing and Intelligence*, vol. 3, no. 1, pp. 38-46, 2011.

[24] T. Mather, S. Kumaraswamy and S. Latif, Cloud Security and Privacy, O'Reilly, ISBN. 978-0-596-80276-9, 2009.

[25] IDP, http://aide.sourceforge.net/.

[26] ac, http://linux.die.net/man/1/ac

[27] accton, http://linux.die.net/man/8/accton

[28] lastcomm, http://linux.die.net/man/1/lastcomm

[29] sa, http://linux.die.net/man/8/sa

[30] iptables, http://linux.die.net/man/8/iptables

[31] NAT, http://www.hjp.at/doc/rfc/rfc1631.html

[32] kpartx, http://linux.die.net/man/8/kpartx

[33] Kernel-based Virtual Machine, http://www.linux-kvm.org/

[34] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin and Anthony Liguori, "kvm: the Linux Virtual Machine Monitor," *In Proceedings of the Linux Symposium*, vol. 1, pp. 225-230, June 2007.

[35] I. Habib, "Virtualization with KVM," *Linux Journal,* Vol. 2008, Feb. 2008.