Future Generation Computer Systems 28 (2012) 379–390

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

CyberGuarder: A virtualization security assurance architecture for green cloud computing

Jianxin Li [a,*], Bo Li [a], Tianyu Wo [a], Chunming Hu [a], Jinpeng Huai [a], Lu Liu [b,**], K.P. Lam [c]

[a] School of Computer Sci. & Eng., Beihang University, Beijing, China
[b] School of Computing and Mathematics, University of Derby, Derby, UK
[c] School of Computing and Mathematics, Keele University, Keele, UK

ARTICLE INFO

Article history:
Received 31 October 2010
Received in revised form
31 March 2011
Accepted 18 April 2011
Available online 11 May 2011

Keywords:
Cloud computing
Green computing
Virtualization
Virtual security appliance
Security isolation

ABSTRACT

As the sizes of IT infrastructure continue to grow, cloud computing is a natural extension of virtualisation technologies that enable scalable management of virtual machines over a plethora of physically connected systems. The so-called virtualisation-based cloud computing paradigm offers a practical approach to green IT/clouds, which emphasise the construction and deployment of scalable, energy-efficient network software applications (*NetApp*) by virtue of improved utilisation of the underlying resources. The latter is typically achieved through increased sharing of hardware and data in a multi-tenant cloud architecture/environment and, as such, accentuates the critical requirement for enhanced security services as an integrated component of the virtual infrastructure management strategy. This paper analyses the key security challenges faced by contemporary green cloud computing environments, and proposes a virtualisation security assurance architecture, *CyberGuarder*, which is designed to address several key security problems within the 'green' cloud computing context. In particular, CyberGuarder provides three different kinds of services; namely, a virtual machine security service, a virtual network security service and a policy based trust management service. Specifically, the proposed virtual machine security service incorporates a number of new techniques which include (1) a VMM-based integrity measurement approach for NetApp trusted loading, (2) a multi-granularity NetApp isolation mechanism to enable OS user isolation, and (3) a dynamic approach to virtual machine and network isolation for multiple NetApp's based on energy-efficiency and security requirements. Secondly, a virtual network security service has been developed successfully to provide an adaptive virtual security appliance deployment in a NetApp execution environment, whereby traditional security services such as IDS and firewalls can be encapsulated as VM images and deployed over a virtual security network in accordance with the practical configuration of the virtualised infrastructure. Thirdly, a security service providing policy based trust management is proposed to facilitate access control to the resources pool and a trust federation mechanism to support/optimise task privacy and cost requirements across multiple resource pools. Preliminary studies of these services have been carried out on our *iVIC* platform, with promising results. As part of our ongoing research in large-scale, energy-efficient/green cloud computing, we are currently developing a virtual laboratory for our campus courses using the virtualisation infrastructure of iVIC, which incorporates the important results and experience of CyberGuarder in a practical context.

1. Introduction

The recent years have witnessed the continuing development of the Internet from its original communication purpose (e.g., email) and content provision (e.g., Web) to an application deployment platform, where increased computing and storage capabilities are constantly being made available to end users. In parallel, an unprecedented number of personal computers are deployed worldwide according to a recent Gartner report, as worldwide PC shipments have reached 82.9 million units just in the second quarter of 2010, representing a 20.7% increase from the second quarter of 2009. At the same, however, enormous energy has been wasted due to idle resources. For example, our own experiments with a Dell Core-2 PC show that it consumes about 85W when sitting idle, corresponding to almost half of the energy when it was fully loaded. A recent report from the NRDC [1] similarly confirmed that most idle servers consume approximately 69%–97% of the

* Corresponding author.
** Corresponding author.
E-mail addresses: lijx@buaa.edu.cn (J. Li), libo@buaa.edu.cn (B. Li), woty@buaa.edu.cn (T. Wo), hucm@act.buaa.edu.cn (C. Hu), huaijp@buaa.edu.cn (J. Huai), l.liu@derby.ac.uk (L. Liu), k.p.lam@cs.keele.ac.uk (K.P. Lam).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.04.012

total power consumption when they are fully loaded, and often when power management function is enabled. With energy costs increasing as the size of IT infrastructures continues to grow, it is apparent that keeping the running costs down is quickly becoming a top priority for many IT centric organisations.

Recently, cloud computing paradigm [2,3] has emerged as an energy efficient approach which enables ubiquitous, on-demand network accesses to a shared pool of flexibly reconfigurable computing resources including networks, servers, storage, applications, and services that can be rapidly deployed with minimal management effort or service provider interactions. In particular, so called virtualisation-based cloud computing platforms are becoming very popular in providing a new supplement, consumption, and delivery model for network software application (*NetApp*) over the Internet. Here, virtualisation refers to the abstraction of computer resources, such as the process of running two or more operating systems on a single set of physical hardware. Originally developed for the IBM mainframe operating systems in the 1960s, the virtualisation technology enables a system administrator to combine disparate physical computing systems into virtual machines in a maximally energy-efficient manner, thus minimizing idle hardware and hence the overall power consumption. Moreover, virtualisation can assist in distributing workload in such a way that servers are either busy, or put in a low power sleep state. This has led to server consolidation, with heightened computer elasticity as well as significantly reduced electricity bills. Based on a software cloud model, a virtualised, scalable and energy-efficient resource management strategy can be developed to facilitate integration of loose-coupled resources, with significantly improved utilisation, and with the added advantage that users can be freed from the often costly administration work including software deployment and maintenance. As with some of the most well known corporations such as Amazon, Google, Microsoft and Salesforce.com, who have become leading cloud service providers, our iVIC[1] platform [4] also utilises a highly virtualised network software operating system to support a elastic, scalable and transparent resource management system for NetApp deployments. More relevantly, it has also been used as a blueprint for constructing energy-efficient or 'green' IT virtualisation infrastructure (viz. IaaS). By leveraging desirable properties of virtual machines (VM) and/or virtual networks, NetApp services can be launched on-demand via the presentation streaming mode using VNC and delivered to PC or mobile phone (e.g., Android) with much improved energy efficiency. Currently, iVIC has been used as an experimental course delivery system for a variety of courses in the undergraduate and postgraduate curricula in Beihang University.

As with most virtualisation-based cloud platforms that operate in a multi-tenant cloud environment, the successful adoption of such a green computing architecture strongly depends on its security assurance mechanisms [5]. Further, it should be noted that, in general, such added computer security services not only impact on the deployment and operations of the system, they are, inevitably, also energy-consuming. Consequently, an integrated security solution would not help in providing the necessary security services, but also in reducing the overall energy consumption. Using our iVIC platform as a testbed, three key challenges have been identified and must be addressed, as follows.

First, a NetApp should be loaded without malicious tempering. Various malwares, such as virus, worms, Trojans and rootkits, continue to threaten the security of a VM. In particular, rootkit malware can hide its own process, or disguise it as a legal process, to escape the detection from a virus scanner or an intrusion detection system (IDS). Fundamentally, the principal problem is that execution of malicious software or codes breaks the integrity of the original system. Several integrity measurements exist, including the well documented approaches of Tripwire [5], IMA [6] PRIMA [7] and Google Chrome OS [8], with widely known limitations. (a) Most of these systems require a significant modification of the operating system (OS) kernel (e.g., the OS kernel in PRIMA needs to be recompiled), and they cannot support legacy applications and 'close-box' operating systems; (b) Many of these systems are developed under the assumption that the OS is fundamentally secure whilst, in practice, most OSs are susceptible to kernel attacks. For example, IMA is implemented through Linux kernel LSM, which is inherently vulnerable to bypass attacks. (c) Some approaches require the support of special hardware; for example, Copilot [9] uses an add-in trusted PCI card to detect modifications to the OS kernel, while Google Chrome OS is based on a solid foundation of Ubuntu. In particular, when a Chrome OS is booted, it first checks the integrity of the OS through TPM to prevent the OS kernel from corruption or tampering by malware. Moreover, different NetApps should have been isolated at different levels when attacks occurred and, as such, security isolation has been the principal approach to counteract such attacks. Here, it is critical for any security services to provide a multi-granularity isolation function, as NetApps can be downloaded from a third party (e.g, AppEngine, AppStore) and may have some malicious codes or faults.[2] Other limitations include: (i) Several procedures were only developed for special business purposes, which lack the generic approach to support NetApp isolation requirements; (ii) The VM isolation cannot be adaptively adjusted according to the secure monitoring status of resource pool; (iii) The firewall network isolation is only a packet filtering mechanism which relies on the physical network connections, and, as a result, cannot be applied to most virtual networks/topologies where shared links may suffer from tapping attacks.

Second, conventional network security systems including IDSs, firewalls, etc should be virtualised and easily deployed in a predominately NetApp execution environment. Here, existing solutions such as *snort*, Cisco *Catalyst* 6500 *W/IDS* hardware have several limitations; namely, (a) The deployment cost of a security system (e.g., IDS) is generally high, and cannot be adaptively redeployed (e.g., Hardware IDS); (b) In place of the traditional security appliance, the virtual security appliance has become a new way of being rapidly encapsulated and dynamically deployed within the distributed IT infrastructure. However, it is difficult to achieve optimal performance with such virtual security appliances, as the underlying physical resource is necessarily shared by several VMs. This problem is particularly aggravated by the deployment of a virtualised network intrusion detection system (NIDS). (c) Virtual security appliances need to handle network traffic fluctuations and frequent network I/Os that inevitably consume a considerably high number of CPU cycles, and would, therefore, necessitate an adaptive mechanism to handle the inherently dynamic nature and requirements of most virtualised architectures.

Third, a policy-based access control service should be put in place to protect the security of the virtual resources. Some NetApps often require scalable computing power, but a single resource pool often found in private clouds may not be able to provide adequate resources for a large number of users. Consequently, multiple collaborating resource pools are often needed to achieve specific goals. For example, the oVirt[3] is built around *libvirt*, which provides additional secure communication (GSSAPI/SASL2) and

[1] http://portal.ivic.org.cn;http://ivic.org.cn/ivic/.

[2] Amazon EC2 provides NetApp isolation with VM, Google AppEngine provides Java, Python binary application isolation, and VMware provides firewall-based network isolation.

[3] http://ovirt.org/.

authentication mechanisms (Kerberos/LDAP) to facilitate access to remote resource pools. In a similar vein, OpenNebula[4] can be used to construct a hybrid cloud that extends a private cloud in order to combine local resources with those made available by remote/public cloud providers such as Amazon EC2 or ElasticHosts. Here, the well known limitations include: (a) Some approaches (e.g., oVirt) only provide a simple identity-based authentication mechanism without considering the real-time security policy updating and evaluation for the multi-tenant resource pool; (b) The existing approaches for the hybrid cloud only provide an interface to invoke other public clouds, and they cannot normally support the federation of multiple resource pools due to potential policy conflict problems for multiple pool federation; (c) The communication channel to remote VMs or virtual networks should, in practice, be secured.

To address these challenges, we propose a novel security assurance architecture, known as CyberGuarder, which enables the trusted loading of NetApps, isolation of different NetApps, virtual security appliances for the NetApp operating environment, and resource access control and remote access to NetApp. Our principal contributions are summarised below:

- We design CyberGuarder, a security assurance architecture designed for NetApp operating systems, as a truly virtualisation-based security solution developed for green cloud computing environments. As proof of the concept, CyberGuarder is also integrated into our iVIC cloud platform that has been constructed as a virtual machine resource management system. Here, CyberGuarder has demonstrated its important security assurance role for the secured operation of iVIC platform.
- A virtual machine security service, which includes mechanisms for software integrity measurement and multi-level security isolation, has also been developed. The Virtual Machine Manager (VMM) based integrity measurement approach named *VMInsight* can provide load-time and run-time monitoring of system processes. VMInsight intercepts system calls and process behaviours by monitoring changes in VM CPU registers. It is implemented in the *hypervisor*, which is completely transparent to the software and operating system running in the VM. Experimental results show that the performance overheads of VMInsight is less than 10% while its additional energy consumption totals to less than 5%. A multi-granularity NetApp sandbox mechanism is also proposed, which can provide both OS users isolation and VMs isolation based on the available tools, including the virtual network isolation solution of ERVIN constructed using a layer-two tunnel VPN between distributed vBridges. Meta-data (e.g. about virtual network topologies) is maintained in a central node to optimise the traffic between VMMs.
- We design a virtual network security service, which provides an adaptive mechanism to deploy virtual security appliances in a virtual network of the NetApps running environment. To enable flexible network traffic detection, we develop a dynamic *software mirror port* mechanism to facilitate detections of the virtual network interface. The mirror port is implemented using an Ethernet bridge configuration tool, *brctl*, which monitors the network traffic. Further, an online controller is also constructed to adaptively control the distributed deployment of vIDS which a security appliance encapsulated the *snort* in accordance with the underlying network topologies, traffic, energy, etc.

[4] http://www.opennebula.org/.

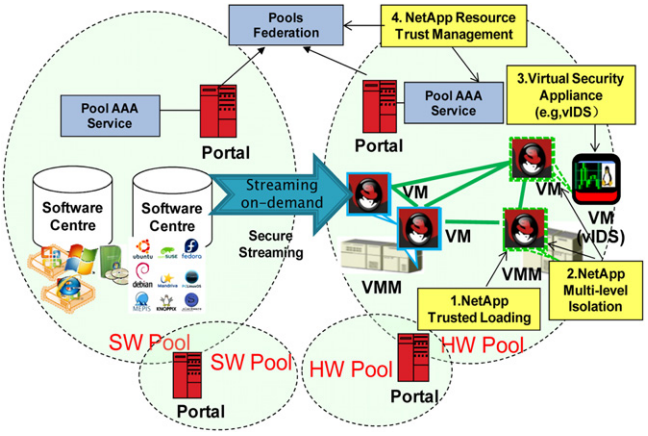


Fig. 1. The deployment architecture of CyberGuarder (AAA: authentication, authorization and accounting. SW: software, HW: hardware, VM: virtual machine, VMM: virtual machine monitor).

- We propose a virtual computing environment security service that offers a policy-based trust management mechanism. The latter not only provides a policy-based access control approach for sharing a resource pool, but also offers a trust federation approach to resource sharing across multiple resources pools. To guarantee the real-time security policy updating and evaluation of the pool resource, we have integrated the policy decision and policy enforcement points into the virtual pool resource information service, and cached the status of the access control list. The trust federation is implemented as an automated authentication procedure using the TrustVO federation policy, allowing users to directly access the VM or virtual network of another pool through secured VNC or VPN clients.

The rest of this paper is organised as follows. Section 2 describes the design of CyberGuarder, with technical details including its performance evaluation results obtained from the iVIC platform. In addition to this section, further related work is described in Section 3. Concluding remarks, including a summary of this paper, are presented in Section 4.

2. Design of CyberGuarder

According to the requirements analysis of a network-based software operating system, we design the architecture of CyberGuarder in iVIC (shown in Fig. 1). iVIC is a network computing platform based on a distributed virtual resource container to encapsulate individual computing and storage devices so that they can provide virtualised entities, such as VMs or vDisks. Virtual machines are dynamically deployed and connected into virtual networks. Users may allocate their own virtual clusters or even complex virtual networks (vLabs) in iVIC to support hardware as a service (vHaaS) and software as a service (vSaaS) application scenario. In iVIC, software and hardware resources are organised in respective resource pools, and software in a software pool (SW Pool) can be downloaded and installed into VMs in a hardware pool (HW pool) on-demand. There are four key security components in this architecture: *NetApp trusted loading*, *multi-level NetApp isolation*, *virtual security appliance* (e.g., vIDS), and *NetApp resource trust management*.

2.1. Virtual machine security service

In this VM security service, we not only provide a VMM-based NetApp trusted loading approach-VMInsight, but also a multi-level security isolation approach based on the virtual machine technologies.

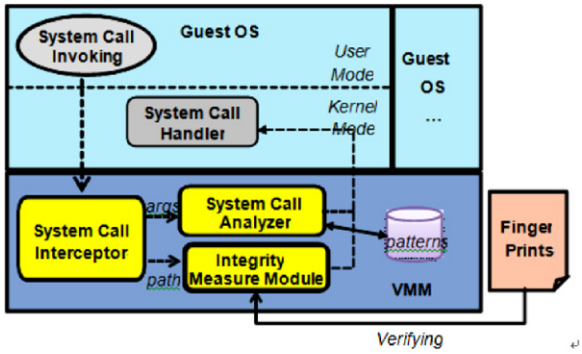


Fig. 2. The architecture of CyberGuarder VMInsight.

2.1.1. VMM-based software integrity verification

In VMImsight, we leverage the VMM-based system call interception approach to provide load-time and run-time integrity protection for a NetApp. Firstly, VMInsight intercepts and analyses the system call sequence to identify and control the loading of software including *user applications*, *shared libraries* and *kernel modules*. Secondly, a system call correlation method is designed to establish the relationship of multiple system calls. Finally, VMInsight monitors the behaviour of NetApp processes to recognise the malicious attacking patterns. For example, VMInsight can find hidden processes using the cross-view theory by collecting a real VM process list and comparing them with that coming from the OS user's tool. The VMM-level protection mechanism ensures that the VM system can maintain its correctness and security even if the guest OS kernel has been comprised. The VMInsight system also supports legacy or commodity guest operating systems, and it requires no modification to the guest OS.

The architecture of VMInsight is illustrated in Fig. 2. VMInsight has three main components: *System Call Interpreter* (SCI), *System Call Analyzer* (SCA) and *Integrity Measure Module* (IMM). The VMInsight works using the following two steps:

(1) SCI intercepts a system call instruction (i.e. INT 80h or sysenter) invoked from the user mode in the guest OS, and identifies a binary-executing related system call, resolves system call arguments to get executable path information.
(2) The arguments and path information are passed to SCA and IMM for further analysis. SCA analyses the system call information based on configurable patterns to monitor the run-time behaviour of processes. IMM receives executable paths from SCI, locates the disk file using path information, and then measures the file content. IMM takes measurements using the SHA-1 hash algorithm, and the fingerprint is then compared with known values stored in the fingerprint library.

Experiments. We have implemented VMInsight in two major VMMs (Qemu and KVM). We have conducted three experiment groups to evaluate its effectiveness to detect malicious processes, and the performance and energy-consumption overhead.

First, we use some malware samples to evaluate the effectiveness of VMInsight. We simulate the malicious software's behaviour of tampering with the already known software to test whether VMInsight can detect such an integrity exception when */usr/bin/ls* is loading. As shown in Fig. 3, VMInsight successfully found the integrity change of */usr/bin/ls*. Next, we test VMInsight's capability of process monitoring using the Apache Web server and some commonly used applications. The results show that VMInsight can identify processes, detect network traffic, and monitor CPU usage and file operations. Such information will be exploited and integrated to identify malicious software behaviours. For example, the hidden processes which steal users' information can be located by analysing the network packet's receive/send status, thus resulting in the reporting of malicious software.

Second, we use some benchmark applications to evaluate the performance overhead of VMInsight for Qemu and KVM. As the results show in Fig. 4, VMInsight incurs less than a 10% performance overhead. According to the above analysis, we can conclude that the monitoring information provided by VMInsight can be used to develop a third-party security system.

Finally, we use four benchmark applications to measure the energy-consumption overhead of VMInsight. We launch four different application tasks as shown in Table 1 on a Dell OptiPlex 960 PC (with Inter(R) Core(TM)2 Quad 2.66 GHz CPU, 4 GB RAM and Debian Linux operating system), and we use Voltech PM1000+ equipment to measure the power and energy consumption of this computer (excluding the monitor). The total energy consumption for each task under different execution environment configuration is listed in Table 2. Based on Table 2, we draw Fig. 5 to show the percentage of energy overhead for KVM and VMInsight compared with the physical machine. From this energy consumption experiment, we can reach two obvious results. First, if we just compare a KVM VM and a VMInsight service with a physical machine, it is obvious that the total consumption energy will increase because both the KVM and security processes will bring some extra overhead to the computer. The results show the overhead incurred by VMInsight is less than 5% on KVM. Second, administrators need to pre-install a security monitor for each VM OS if there is no a virtualisation layer security service. It will not only incur a heavy management burden, but also the utilisation of the security monitor software is often very low, so computer resources and energy are wasted. In particular, the energy consumption overhead incurred by the monitor for every OS is generally 5%. If a security monitor is installed on each VM OS, the total energy wasted in a physical machine will significantly increase because a physical machine can generally run 20–30 Linux VMs. However, the VMInsight is only a module on the VMM layer, and it can serve all VMs.

PID	PPID	NAME	CPU%	SendBytes	RecvBytes	ReadBytes	WriteBytes	PATH	MD5	Trusted
1581	1573	rsyslogd	0%	12288	0	0	0	/usr/sbin/rsyslogd	6678abfd8d3d8a2a97bca3e0fcb4f984	YES
1593	1592	acpid	0%	16384	5	0	0	/usr/sbin/acpid	c25682102e4fe0d345923b55134939d5	YES
1613	1612	cron	0%	16384	0	76	0	/usr/sbin/cron	7149651c7dbc672e23f1ae0e655cf8f2	YES
1631	1626	apache2	0%	36864	104	0	0	/usr/sbin/apache2	066bc91b46a38490bf42b15c6c2ca454	YES
1632	1631	apache2	0%	0	0	0	0	/usr/sbin/apache2	066bc91b46a38490bf42b15c6c2ca454	YES
1752	1683	login	0%	18063	177	0	0	/bin/login	1e6966654edb9c8c106b989c55bc324a	YES
1763	1761	apt-get	89%	162977040	16863513	131	2048	/usr/bin/apt-get	141b163399273613559a799a20cfd3c5	YES
2839	2802	mysqld_safe	0%	95448	1008	0	0	/usr/bin/mysqld_safe	849762f87aa01e1af972b6a8205f8a17	YES
2845	1752	ls	0%	0	0	0	0	/usr/bin/ls	573ec4e8d3095cc33106c62cced6fc9e	NO

Fig. 3. System processes monitored by VMInsight.

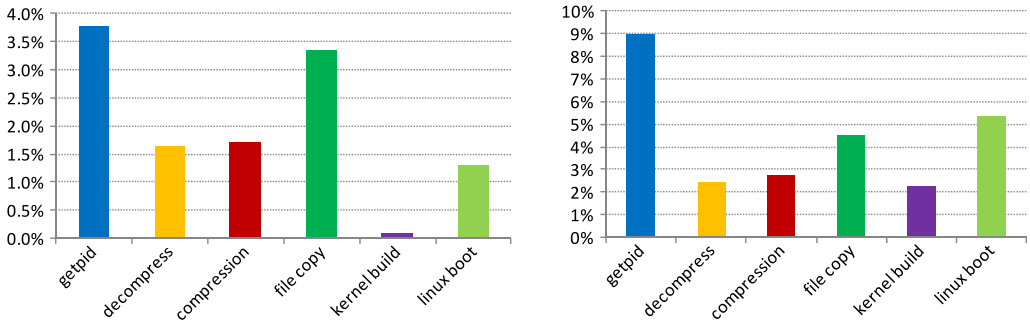


Fig. 4. Runtime overhead of VMInsight on Qemu (on the left) and KVM (on the right).

Table 1
The four experiment applications' tasks.

Application task	Command string
Kernel build	make defconfig & make
File copy	cp −r linux-source-dir otherwhere
Compression (gz)	tar zcf linux-source-dir
Decompression (bz2)	tar zxf linux-source.tar.bz2

Table 2
The total energy wasted for each task (W s).

Application task	Physical machine	KVM VM	KVM with cyberGuarder VMInsight
Kernel build	197×10^3	295×10^3	304×10^3
File copy	2.9×10^3	3.7×10^3	3.9×10^3
Compression (gz)	3.1×10^3	3.6×10^3	3.7×10^3
Decompression (bz2)	3.2×10^3	4.0×10^3	4.1×10^3

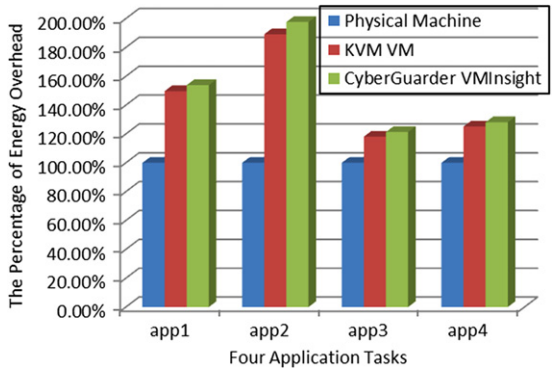


Fig. 5. The energy consumption overhead for different tasks.

2.1.2. Multi-granularity NetApp sandbox mechanism

Isolation is an important factor to improve the availability and security of applications running in a virtual environment, and is far superior to applications running in a traditional, non-virtualised system. While virtual machines can share the physical resources of a single computer, they remain completely isolated from each other as if they were in separated physical machines. If, for example, there are four virtual machines on a single physical server and one of the virtual machines crashes, the other three virtual machines remain available.

As shown in Fig. 5, we design a multi-granularity NetApp sandbox mechanism in CyberGuarder, which can provide security isolation at different levels. The isolation at the user and application levels is achieved with existing tools, and the virtual network level isolation is achieved with CyberGuarder ERVIN. In response to the user isolation requirement in an OS, we use *chroot* to create and host a separate virtualised copy of the software system. Now, we are also adopting Linux kernel *seccomp* to allow processes to call a very small subset of system calls, e.g., read, write, sigreturn, and exit. For the NetApps isolation requirement among VMs, we assign security policies for a resource pool and a scheduler (deployed with the Web Portal) can automatically deploy VMs according to the NetApps' isolation policies. For the virtual network isolation requirements, we design the ERVIN which uses a layer-two tunnel VPN between distributed vBridges, and the meta-data such as virtual network topologies is maintained in a central node to optimise the traffic between VMMs. CyberGuarder ERVIN provides a data transmission mechanism in a P2P manner for a virtual network, the network packets between different virtual machines do not transit through a central server, thus making full use of the network bandwidth between the hosts to improve the efficiency of virtual machines' network packet transmission (Fig. 6).

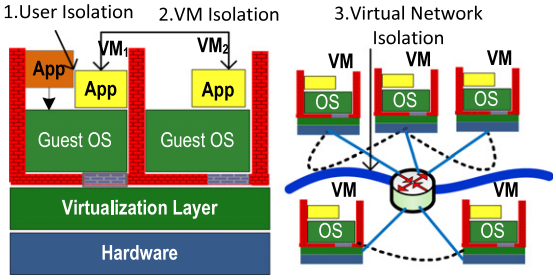


Fig. 6. Three-level NetApp isolation in CyberGuarder.

Experiment. In order to evaluate the performance of CyberGuarder, ERVIN can take advantage of the network bandwidth between hosts compared with the OpenVPN-based virtual network approach. We design an experiment to measure their performance. We test the performance of virtual networks connected with 2, 6 and 10 virtual machines respectively, and each virtual machine is deployed on different hosts, and we use NetPIPE to measure the network throughput. In Fig. 7, we plot the network throughput against the size of packet sent between VMs by NetPIPE. As shown in Fig. 7, CyberGuarder ERVIN has better performance compared with OpenVPN on both communication throughput and scalability. There is no obvious performance degradation when increasing the number of virtual network peers (because the experimental virtual machines lie on different hosts), while the OpenVPN performance is greatly affected by the scale of virtual network and communication overhead. It drops to 48% of the maximum speed when the number goes up to 5.

2.2. Virtual network security service

In CyberGuarder, we also design an adaptive security system deployment mechanism for virtual network environments. We encapsulate traditional network security systems into virtual security appliances and adaptively deploy them into virtual networks to safeguard the applications running in the virtual

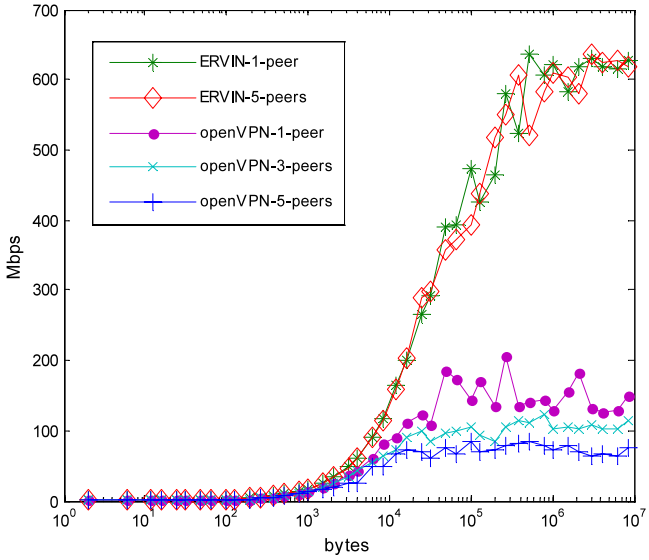


Fig. 7. The performance of CyberGuarder ERVIN vs. openVPN.

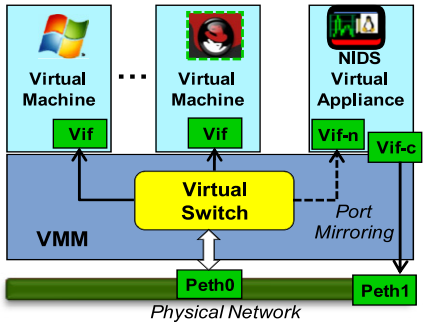


Fig. 8. The architecture of CyberGuarder vIDS.

networks. We also design a dynamic provision approach based on fuzzy control theory, which can continuously control resource allocation for virtual security appliance to deal with varying network traffic while still satisfying the performance or energy consumption requirements.

As shown in Fig. 8, VMs Vifs, Peth0 and the Vif-n of vIDS are connected with a virtual switch. Peth0 is a physical network interface, and all packets of a VM will go through Peth0. Virtual NIDS has two virtual network interfaces: Vif-n and Vif-c. The Vif-n is connected with the mirror port of the virtual switch, and duplicates and forwards monitored packets to the vIDS. To guarantee no disturbance to the whole system, a physical network interface Peth1 is dedicated for Vif-c to connect with a physical network. Linux bridge works in the layer 2 protocol, and acts in a similar manner with physical switch, so we choose Linux bridge as a virtual switch in our implementation. We have a slightly modified Linux bridge to support port mirroring, where a flag is added to net_bridge_port struct to indicate whether the network traffic traversing this port will be duplicated and forwarded or not, and a pointer is added to net_bridge struct, it points to a bridge port to indicate that this port is the mirror port of the bridge. This is a very flexible approach to integrate with available network intrusion detection systems. Any port can become a mirror port. We can dynamically configure which virtual network interface is under detection and which is not. We have added four commands to brctl,[5] add_mirror_port, del_mirror_port, add_src_if, and del_src_if.

[5] A user-mode tool for controlling Linux Bridge.

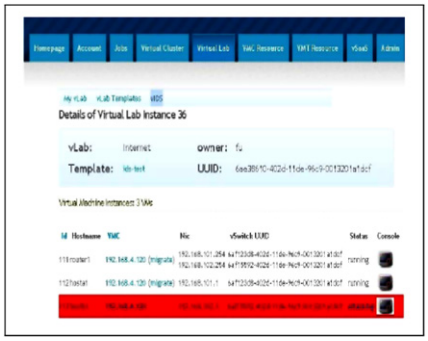


Fig. 9. A snapshot of vIDS demo in iVIC.

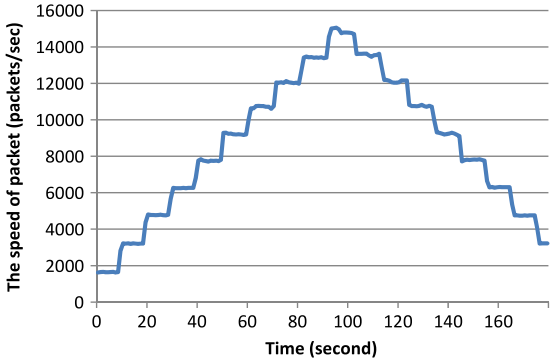


Fig. 10. The simulated workload for vIDS.

To enable the port mirroring function, we first need to execute "brctl add_mirror_port ⟨ifname⟩" to assign a bridge port to be the mirror port, and any packet forwarded to the mirror port will be sent to the virtual network interface connected with this port. Next, we call "brctl add_src_if ⟨src_if_name⟩" to specify that the packets flowing through the "src_if_name" interface will be duplicated and forwarded to the mirror port. If we want to cancel the monitoring of one interface, we can run the command "brctl del_src_if ⟨ifname⟩". Finally, we use "brctl del_mirror_port ⟨ifname⟩" to turn off the port mirroring function.

Fig. 9 also shows a snapshot of our vIDS demo in iVIC portal. The iVIC Portal is deployed as the user interface and provide management console for a virtual machine pool. In iVIC Portal, users can create a virtual cluster or virtual lab with some complex virtual networks. In this figure, a virtual lab instance is created with vIDS service, and if a virtual machine is detected be attacked, the attacked virtual machine will be highlighted in red.

Experiment. We have implemented CyberGuarder vIDS which is based on snort. To evaluate its packet analysis performance and the power changing with the time-varying workloads, we have conducted two experiments. Fig. 10 shows the simulated workload of 160 s, we change the packets sending speeds every 10 s.

We launch vIDS to evaluate the effectiveness of our dynamic provision approach. Fig. 11 shows the transient and accumulated drop rate for 2% MPDR (Maximum Packet Drop Rate). We can see that the transient drop rate fluctuates up or down at the MPDR, while the accumulated drop rate tends to gradually converge at the MPDR. The results show that our system can precisely allocate resource for NIDS according to its resource demands, while still satisfying the performance requirements of NIDS.

At the same time, we also use Voltech PM1000+ equipment to measure the power and energy consumption of this computer (excluding the monitor). From the Fig. 12, we can see the power of CyberGuarder vIDS (encapsulating a snort IDS) quickly reduces with the decreasing of network workload, but the power

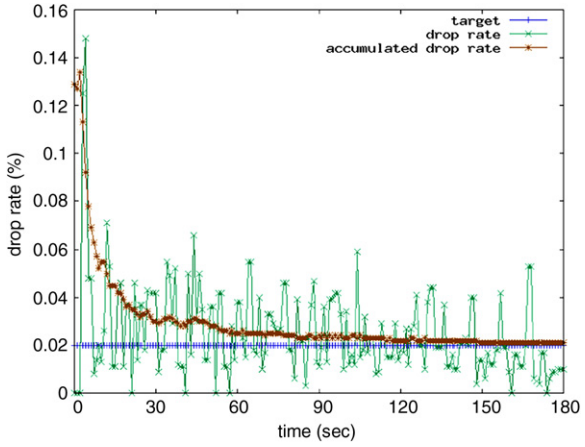


Fig. 11. Transient and accumulated packet drop rate for 2% maximum packet drop rate.

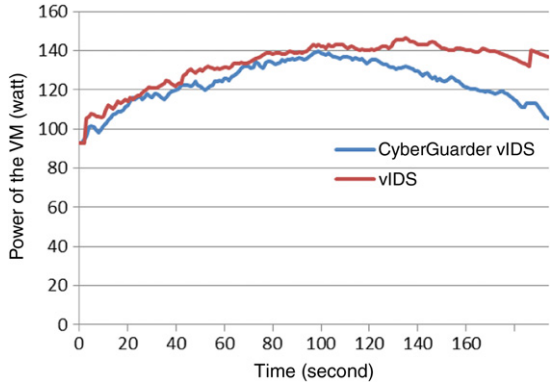


Fig. 12. The power of CyberGuarder vIDS and common vIDS.

of common vIDS reduces very slowly. This is because that CyberGuarder vIDS has the capability to dynamically control the CPU usage based on the workload of packet receiving, but the common vIDS does not.

2.3. Virtual environment security service

- **Policy-based security service for the local VM pool**

In a VM pool, many physical machines are connected in a high-speed network, and every physical machine can run several virtual machines simultaneously. In this pool, users can create their own virtual clusters or virtual labs by connecting assigned virtual machines. The security policy for this virtual machine pool is configured in a centralised portal, and it authenticates the user's identity, and manages the access control policies of virtual machines.

As shown in Fig. 10, the steps of security policy enforcement in a VM pool are as follows: (1) The user firstly logs into the virtual machine pool with their password or certificate; (2) The authentication server in the VM pool verifies the identity of the login user; (3) The user creates a virtual cluster or a virtual lab on the portal workspace. (4) When the user performs a task involving some operations on the virtual machine pool, these actions need to be authorised firstly by the user policy server. If all the actions involved in this task are permitted, then this task is submitted to the scheduler. (5) The portal submits a description file of the user's task to the scheduler. (6) The scheduler deploys the virtual machines according to the description of the task file and pool information service. After a virtual cluster or a virtual lab is deployed in the VM pool, then the user can access directly

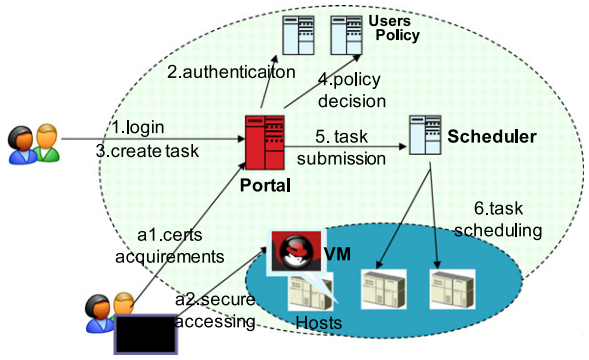


Fig. 13. The architecture of security policy enforcement in a local VM pool.

Table 3
The operations for VM pool and physical machine.

Object	Operation	Statement
VM pool or physical machine	Any	A wildcard that represents any operation on the VM pool
VM pool	ivic#createVCluster (VMCount)	Create a virtual cluster with a variable that specifies the number of virtual machines in this cluster
VM pool	ivic #createVLab VMCount, (VSwitchCount)	Create a virtual lab with two variable that specify the number of virtual machines and the number of virtual switches in this cluster
Physical machine	ivic #deployVM	The operation to deploy a virtual machine into the VM pool
Physical machine	ivic #deploySwitch	The operation to deploy a virtual switch into the VM pool
Physical machine	ivic #startVM	The operation to start a virtual machine deployed in the physical machine
Physical machine	ivic #startSwitch	The operation to start a virtual switch deployed in the physical machine

Table 4
The constrain variables with the operations for VM pool and physical machine.

Constraints variable	Statement
vLabCount	The maximum number of virtual labs that user can create
vClusterCount	The maximum number of virtual clusters that user can create
vSwitchCount	The maximum number of virtual switches that user can create
liveSwitchCount	The maximum number of virtual switches that a physical machine can run
liveVMCount	The maximum number of virtual machines that a physical machine can run

related virtual machines in this pool via remote client tools. The procedures are: (a1) The user firstly requests a proxy credential or certificate from the portal after an automated authentication procedure; (a2) The user can access the virtual machine through SSH or VNC clients (Fig. 13).

On the side of policy server, the policy is stored with format of *policy* = (subject, object, constrains set), where the subject is a user, the object can be a VM pool or a physical machine, and the constraints set includes operation constraints on the VM pool and physical machines. The possible operations for a VM pool or physical machines are listed in Table 3, and the constraint variables with the operations are listed in Table 4.

An example of the security policy is as follows:
(Alice@ivic.org.cn, pool-1, [vClusterCount $\leq$ 2]);
(Alice@ivic.org.cn, Host: 192.168.0.119, [liveVMCount $\leq$ 2]).

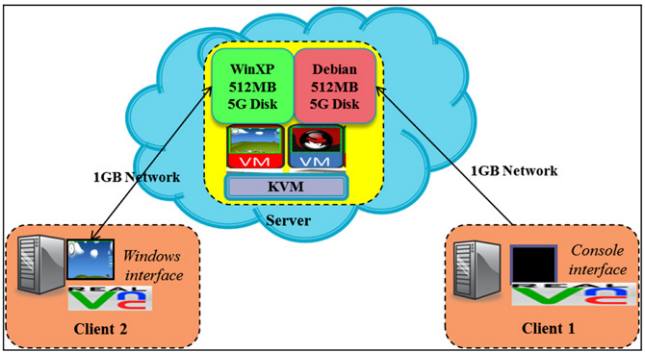


Fig. 14. Remote VM access through VNC with TLS and without TLS.

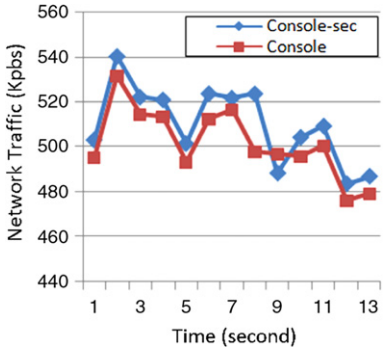


Fig. 15. Network traffic of communication using console client with TLS and without TLS.

This policy means that Alice mostly can create two virtual clusters on VM pool-1, and specially can mostly start two virtual machines on the physical machine 192.168.0.119 simultaneously.

Experiment. Because the cloud is a centralised infrastructure, all VMs are executed on the server side. If a client wants to interact with its VM, it must connect this VM based on remote display tool (e.g., VNC used in CyberGuarder). However, if too many clients connect their VMs located in a resource pool, the bandwidth will become an issue. Therefore, our security mechanism should guarantee not to bring too much traffic overhead. In this experiment, we measure the network traffic. As shown in Fig. 14, we configured a test environment on a QEMU VNC Server for two virtual machines and a RealVNC viewer, with a 1 GB network connection. Then, we measure the communication throughput when the channel from VNC desktop to the VNC server is encrypted or not. On the test server physical machine, the network interface cards of virtual machines is bridged to the physical network of a computer. The experiments are divided into two groups, one runs Debian Linux on a virtual machine with a console, and a simple 'ls' shell command is executed to continually refresh the screen, and another runs Windows XP OS on a virtual machine with a Windows desktop, and the Media Player is launched to play a video in full-screen mode on a desktop of 1024 × 768 resolution. On the client side, we access these two virtual machines through a RealVNC client with TLS encryption and without encryption respectively.

The results are shown in Figs. 15 and 16. Fig. 15 indicates that average network traffic is about 500 kbps for a console client, and the security mechanism only adds a little network traffic. The Fig. 16 indicates that the two modes almost have the same network traffic (about 2000 kbps) because the encryption brings a lower percentage to the larger total network traffic when a Windows desktop is transferred.

- **Policy-based trust federation for multiple VM pools**

In general, a company or organization can build its own private resource pool (private cloud), while the resource capacity of a private pool cannot fulfill all business requirements. For example, the required computing resources of an application in Facebook increased to 3000 hosts from 50 hosts in only three days. In CyberGuarder, we design a policy-based trust management service named TrustVO (shown in Fig. 17) to improve its scalability by federating multiple resource pools (clouds). The security policy is specified by role mapping, and possible conflicts are resolved by our existing work PEACE-VO [10].

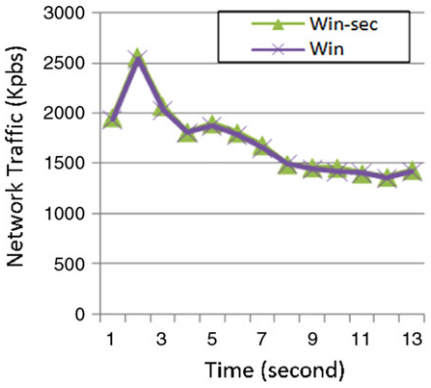


Fig. 16. Network traffic of communication using Window client with TLS and without TLS.

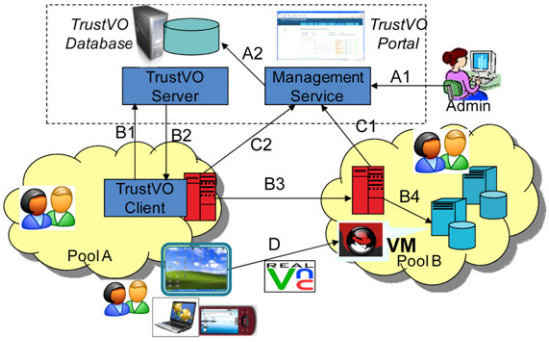


Fig. 17. The management and authentication workflow of CyberGuarder TrustVO.

If some pools need to be federated, the management steps of TrustVO administrator are as follows:

- A1: The *Admin* logins into the *TrustVO Portal*.
- A2: The *TrustVO Server* authenticates the identity of *Admin* and executes the management for *VO Database* via *Management Service*.

When a client from *Pool A* wants to use resources provided in *Pool B*, the authentication steps of TrustVO are as follows:

- B1: The *Pool A*sends VO membership credential requests to the *TrustVO Server*.
- B2: The *TrustVO Server* generates a credential according to the requests.
- B3: The *Pool A Portal* sends a job scheduling request to another *Pool B Portal*.
- B4: The *Pool B* makes authorization decision for the job scheduling across pools (clouds).
- C1, C2: The information of federated virtual resource is reported to the *Management Server*, and clients can query the resource's status through accessing the *Management Server*.
- D: The user can access any authorised VM Desktop via RealVNC.

Table 5
Authentication time with different security communication mechanisms.

	SOAP security (WS-security)	OpenSSH RSA
Authentication time (ms)	350	380

Experiment. To provide a general trust federation approach to interoperate with the existing cloud security infrastructures. In CyberGuarder TrustVO, an administrator can configure different security communication mechanisms. Currently, CyberGuarder can support two major security communication toolkits, GSI SOAP and OpenSSH. The SOAP security mechanism has been extensively used in Grid security infrastructure and some SOA security services. The OpenSSH is also a traditional remote secure access tool and can be easily used. We use these two security communication mechanisms for TrustVO respectively. As shown in Table 5, the average authentication time is about 350 ms for SOAP security and 380 ms for the OpenSSH RSA mechanisms, and there are no obvious differences between these two mechanisms.

3. Related work

In 1970s, Madnick and Donovan [11] from MIT, who had engaged in research work relevant to IBM VM/370, firstly put forward the idea to improve the system security based on a virtual machine isolation mechanism. Many years later, the virtualisation technology began to receive attentions again with the prevalence of new Internet-based computing paradigms, e.g., Cloud computing.

3.1. Virtualisation security

In 2008, Borders et al. from Michigan University summarised some security technologies related to virtualisation. This is an earlier overview and analysis to the security mechanisms of the virtual machine [12], and it introduces related work including intrusion detection, intrusion defense and the honeypot system based on virtualisation technology and so on. Nowadays, the virtualisation technology has been brought to the forefront in the area of industry and manufactory, and it has also become an important technology to build a green IT infrastructure. However, the question remains how to ensure its security becomes a bottleneck of its real adoption. In 2009, 11 scholars from UC Berkeley Reliable Adaptive Distributed Systems Laboratory published a report on cloud computing [13]. In this report, they give a concept model and some research trends of cloud computing. In particular, the top 10 problems relating to cloud computing are discussed, and three of them are related to the security issues. To build a secure execution environment for the network software applications, the related work on virtualisation can be classified into three types: security isolation, trusted loading, and monitoring & detection.

Security isolation

In nearly 40 years of virtualisation technologies, virtual machines have been used from the previous physical environment isolation to dynamic business logic isolation, and virtualisation computing systems have realised the balance and synthesis of multiple functions, such as computing performance, application efficiency, security isolation and so on. At the same time, we are suffering from more system vulnerabilities and network attacks which frequently occur. An important motivation of early IBM VM/370's appearance is to realise partition isolation, and the VMM (Virtual Machine Monitor) technology makes it possible to create many of virtual machines to run independent operating systems in the same physical hardware. The virtualisation system can avoid the system information leakage, which may be caused by users' improper or malicious operations. Yan Wen [14] has presented a virtualisation isolation model, and he put forward a new kind of virtual machine model based on the hardware abstraction layer virtualisation (Safe Virtual Execution Environment, SVEE), which implements the Bell–LaPadula confidentiality model and Biba integrity model. Researchers from MIT provide the Flume system [14] which is used for distributed information flow control, managing the data flow of application segments and realizing the integrity and privacy of data. However, the cloud is generally a multi-tenant computing environment, and an isolation solution at different levels is required. In CyberGuarder, we achieve this goal through lightweight application isolation, virtual machine isolation and virtual network isolation.

Trusted loading

A fundamental reason for an unreliable system is because the integrity of systems is broken by malicious software or codes. Therefore, ensuring the software originates from a trusted party is an effective way to guarantee the intrinsic security of a system. The integrity measurement is a way to prove that the providers and sources are reliable and accountable, meaning the software files have not been damaged or tampered with. The researchers from IBM propose the HIMA [15] which also employs a VMM-based approach to take integrity measurements of user programs and kernel codes. However, HIMA needs modification to the guest OS. Arvind Seshadri et al. from CyLab of CMU designed a lightweight hypervisor named SecVisor which can ensure the integrity of the kernel code of Linux and prevent malicious code injection and so on. In a word, the major method of SecVisor is to virtualise the MMU and IOMMU. However, SecVisor is a lightweight hypervisor, and it can only operate one guest OS and cannot be used when there are hybrid memory pages.

Security monitoring and detection

Monitoring technology is another key way to keep the system running healthily. The VMM has a good introspection capability, thereby it's intensively used to monitor the device status and attack behaviours. The research work on this direction can be classified into two types: one is a pure monitor function for the memory, disk and I/O of virtual machines, and another is the system security detection including the malicious attacks and some intrusion behaviours. Payne and Wenke Lee proposed the XenAccess library [16] which was based on the Xen 3.0's existing XenControl library and Blktap arch. XenAccess mainly aims to monitor the virtual machine memory and disk I/O, and this approach could be easily extended to monitor network flow and CPU, but XenAccess needs to be deployed to Xen Domain0. Payne et al. further proposed Lares [17] which realised an active monitoring function based on Xen and Window XP. Garfinkel [18] from Stanford University studied virtualisation's characteristics such as mobility, security monitoring etc., and he proposed some approaches for intrusion detection, integrity checking and a honeypot system and forensics based on the monitoring capability of the VMM layer. For example, a VMM introspection based architecture for intrusion detection is used to analyse the attackers' behaviour. KvmSec [19] is an extension of Linux Kernel Virtual Machine (KVM), which can prevent KVM from being attacked by virus and kernel rootkit. KvmSec provides a transparent way to data collection and analysis to the guest OS. J. Ashlesha et al. [20] from Michigan University propose a detecting past and present an intrusions method through vulnerability specific predicates. This method can monitor the internal running state of a virtual machine based on the introspection capability.

Besides, many IT companies such as Amazon, Google, and Microsoft have launched their cloud computing and green computing projects, and the virtualisation security is also a major product.

In the Amazon S3 storage service, the owner can assign access control policy to specify who can read or write or have other privileges. In the Amazon EC2 computing services, unauthorised access to the virtual machine or virtual network can be prevented through a firewall policy configuration on IP and routing. VMWare VirtualCenter is a kind of task-based privilege management system, which is used to control the permissions of administrators and users on the platform. The system administrators can assign the user permissions through configuration of user/group, roles to tasks. VMWare vSphere is a cloud operating system including VMsafe and VMWare vShield Zones, and they provide firewall, anti-virus, intrusion detection and intrusion prevention capabilities to the virtual environment. VMware vShield Zones can configure VLAN to separate the network and create a security boundary. Microsoft Hyper-V provides some security functions based on a virtual machine, such as malicious code execution prevention, role-based access control, and streamlined system architecture. RedHat oVirt integrates the user and data management based on LDAP, and distributes the user ticket based on Kerberos infrastructure, and uses freeIPA project to implement virtual resource authentication, authorization and accounting. CyberGuarder is a much different solution compared with these products, and three kinds of security assurance services on the granularity of virtual machine, virtual network and virtual pool, and the security mechanisms can be smoothly integrated with the virtualisation infrastructure and interoperated with the existing local security infrastructures.

3.2. Green cloud computing

The power consumption of computers and data centers is growing at an unprecedented level: the EPA estimates US data centers will consume 100 billion kilowatt hours in 2011. Much of this energy is wasted in idle systems: in typical deployments, server utilisation is below 30%, but idle servers still consume 60% of their peak power draw. In recent years, there have been many research efforts focusing on how to achieve energy efficiency for computers. The traditional energy saving approaches for a computer include CPU and storage equipment improvements and power management and dynamic voltage scaling (DVS) technologies based on operating systems. For instance, David Meisner et al. propose the PowerNap [21], which is an energy-conservation approach where the entire system transitions rapidly between a high-performance active state and a near-zero-power idle state in response to instantaneous load, and a power provisioning approach provides high conversion efficiency across the entire range of PowerNap's power demands. Kephart et al., [22] build a framework with consideration to both the power management and performance management, and use two existing IBM products, one that manages performance and one that manages power through dynamic voltage scaling (DVS) approach, resulting in power savings of approximately 10%.

The traditional energy-saving approaches are mainly based on operating systems with a full knowledge of and full control over the underlying hardware, but the distributed nature of multi-layered virtual machine environments makes such approaches insufficient. Cloud virtualisation can significantly improve efficiency by leveraging the utilisation and consolidation of virtual machines with a minimum number of powered physical machines. Obtaining energy efficiencies in data centers is highly specialised and capital intensive. In 2009, Francis and Richardson [23] presented a green maturity model for virtualisation, and focused on a reduction in energy consumption over the full equipment life cycle as the prime motivator for "green" application design. Some researchers [24] from IBM have presented a server workload analysis for power minimization user consolidation to reduce the datacenter's energy. The basic principal is turning on/off the server according to specific policies. Stoess et al. [25] present a novel framework for energy management in modular, multi-layered operating system structures. This framework targets hypervisor- based virtual machine systems, and the guest level energy management relies on effective virtualisation of physical energy effects provided by the VMM.

Virtualisation technologies have been extensively studied, and security services should be provided. However, they generally will add some extra energy consumption, and there are few related works considering this issue [26]. CyberGuarder designed in this paper is an important enhancement to the security of a green cloud computing environment. First, security is an important foundation to enable the green cloud computing infrastructure to be actually deployed and applied. Second, the CyberGuarder itself provides a security service based on virtualisation technology in different grained levels with the many energy-saving benefits of virtualisation. For example, some security service is deployed in virtual machines, and it can also be dynamically deployed or consolidated and can sleep or shut down when the risk is low. Finally, we also provide some energy-aware approaches to the security services policy, thereby administrators can dynamically deploy or control the security services according to the energy or performance requirements.

4. Conclusion

Modern computing presents not only technical problems, but also major environmental challenges in terms of its high energy consumption. As the sizes of IT infrastructure continue to grow, it is clear that effective green IT solutions must be developed to minimise its impacts on our environment. Here, cloud computing offers a natural extension of virtualisation technologies which enable scalable management of virtual machines residing on distributed hosts, thus allowing maximal utilisation of the underlying re-sources and replacing the traditional "one server, one application" model with a multi-tenant architecture/model of cloud services. As well as providing a much improved hardware efficiency (through sharing), the latter also raises serious issues of security and data privacy within the cloud environment in general, and in an open NetApp operating system in particular. This paper proposes a virtualisation security assurance architecture, CyberGuarder, which is designed to address several key security problems within the "green" cloud computing context. In particular, CyberGuarder provides three different kinds of services; namely, a virtual machine security service, a virtual network security service and a policy based trust management service. Several techniques concerning the provision of integrity verification, multi-level NetApp isolation, virtual security appliance (e.g., vIDS), and NetApp resource trust management services have been discussed in some detail. Preliminary results obtained on our iVIC platform are promising; for example, the energy consumption of CyberGuarder is significantly lower than the most widely used IDS of Snort, as CyberGuarder's vIDS can dynamically adapt the CPU utilisation to match workload. At present, we are incorporating these results and our experience with CyberGuarder in the practical development of iVIC, a virtualisation infrastructure for an experimental course delivery system for both undergraduates and postgraduates in Beihang University. Future work concerns the construction of a reliable and scalable NetApp operating system that will support, through advanced virtualisation technologies, a greater federation of cloud services to facilitate the seamless integration of private and public cloud systems. This warrants further active research.

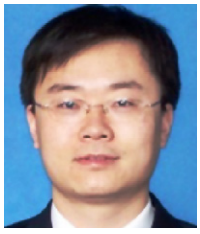
Acknowledgment

The authors gratefully acknowledge the anonymous reviewers for their helpful suggestions and comments. This work is partially supported by Program for New Century Excellent Talents in University 2010 and the Fundamental Research Funds for

the Central Universities, National Nature Science Foundation of China (No. 60903149, 91018004), China 863 Program (No. 2009AA01Z419), and China 973 Fundamental R&D Program (No. 2011CB302602, 2011CB302603).

References

[1] Natural Resources Defense Council "Recommendations for Tier I ENERGY STAR Computer Specification", http://www.energystar.gov/ia/partners/prod_development/revisions/downloads/computer/RecommendationsTierICompSpecs.pdf.
[2] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 599–616. doi:10.1016/j.future.2008.12.001.
[3] Gabriel Mateescu, Wolfgang Gentzsch, Calvin J. Ribbens, Hybrid computing-where HPC meets grid and cloud computing, Future Generation Computer Systems 27 (5) (2011) 440–453. doi:10.1016/j.future.2010.11.003.
[4] Yang Chen, Tianyu Wo, Jianxin Li, An efficient resource management system for on-line virtual cluster provision, cloud, 2009, in: IEEE International Conference on Cloud Computing, 2009, pp. 72–79.
[5] Dimitrios Zissis, Dimitrios Lekkas, Addressing cloud computing security issues, Future Generation Computer Systems (2011) in press (doi:10.1016/j.future.2010.12.006).
[6] R. Sailer, X. Zhang, T. Jaeger, L. van Doorn, Design and implementation of a tcg-based integrity measurement architecture, in: Proceedings of the 13th USENIX Security Symposium, August 2004.
[7] T. Jaeger, R. Sailer, U. Shankar, Prima: policy-reduced integrity measurement architecture, in: Proceedings of the 2007 ACM Workshop on Scalable Trusted Computing, SACMAT'06, June 2006.
[8] Google Chrome OS. www.chromium.org/chromium-os.
[9] N.L. Petroni Jr., T. Fraser, J. Molina, W.A. Arbaugh, Copilot — a coprocessor-based Kernel runtime integrity monitor, in: Proceedings of the 13th USENIX Security Symposium, 2004.
[10] Jianxin Li, Jinpeng Huai, Chunming Hu, Yanming Zhu, A Secure Collaboration Service for Dynamic Virtual Organizations, in: Information Sciences, vol. 180, Elsevier, 2010, pp. 3086–3107.
[11] E.M. Stuart, J.D. John, Application and analysis of the virtual machine approach to information system security and isolation, in: Proceedings of the Workshop on Virtual Computer Systems, ACM, Cambridge, Massachusetts, United States, 1973.
[12] X. Zhao, K. Borders, A. Prakash, Virtual machine security systems, book chapter, Advances in Computer Science and Engineering (2009) 339–365. http://www.eecs.umich.edu/~aprakash/eecs588/handouts/virtualmachinesecurity.pdf.
[13] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, Above clouds: A Berkeley view cloud computing, Technical Report No. UCB/EECS- 2009-28, University California, Berkeley, 2009. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf.
[14] M. IKrohn, A. Yip, M. Brodsky, N. Cliffer, M.F. Kaashoek, E. Kohler, R. Morris, Information flow control for standard OS abstractions, in: Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles, Stevenson, Washington, USA, October 14–17, 2007, SOSP'07, ACM, New York, NY, 2007, pp. 321–334. doi:10.1145/1294261.1294293.
[15] Ahmed M. Azab, Peng Ning, Emre C. Sezer, Xiaolan Zhang, HIMA: a hypervisor-based integrity measurement agent, in: Proceedings of the 25th Annual Computer Security Applications Conference, ACSAC'09, December 2009, Honolulu, Hawaii, USA.
[16] B.D. Payne, W. Lee, Secure and flexible monitoring of virtual machines, Presented at 23rd Annual Computer Security Applications Conference, ACSAC, Miami Beach, Florida, USA, 2007.
[17] B.D. Payne, M. Carbone, M.I.S.W. Lee, Lares: an architecture for secure active monitoring using virtualization, Presented at IEEE Symposium on Security and Privacy, S&P 2008, Oakland, California, USA, 2008.
[18] T. Garfinkel, M. Rosenblum, When virtual is harder than real: security challenges in virtual machine based computing environments, Presented at Proceedings of the 10th Conference on Hot Topics in Operating Systems, HOTOS 2005, Berkeley, CA, USA, 2005.
[19] L. Flavio, P. Roberto Di, KvmSec: a security extension for Linux kernel virtual machines, in: Proceedings of the 2009 ACM Symposium on Applied Computing, ACM, Honolulu, Hawaii, 2009.
[20] J. Ashlesha, T.K. Samuel, W.D. George, M.C. Peter, Detecting past and present intrusions through vulnerability-specific predicates, in: Proceedings of the Twentieth ACM Symposium on Operating Systems Principles, ACM, Brighton, United Kingdom, 2005.
[21] David Meisner, Brian T. Gold, Thomas F. Wenisch, PowerNap: eliminating server idle power, in: Proceeding of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS'09, ACM, New York, NY, USA, 2009, pp. 205–216. doi:10.1145/1508244.1508269.
[22] Kephart, J. Chan, H. Levine, D. Tesauro, G. Rawson, F. Lefurgy, C, Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs, in: Proc. IEEE Intl. Conf. on Autonomic Computing, ICAC, Jun. 2007, pp. 145–154.
[23] Kevin Francis, Peter Richardson, Green Maturity Model for Virtualization, http://msdn.microsoft.com/en-us/library/dd393310.aspx.
[24] Akshat Verma, Gargi Dasgupta, Tapan Kumar Nayak, Pradipta De, Ravi Kothari, Server workload analysis for power minimization using consolidation. in: Proceedings of the 2009 Conference on USENIX Annual Technical Conference, USENIX'09, USENIX Association, Berkeley, CA, USA, 2009, pp. 28–28.
[25] Jan Stoess, Christian Lang, Frank Bellosa, Energy management for hypervisor-based virtual machines. in: Jeff Chase, Srinivasan Seshan, (Eds.), 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference ATC'07, USENIX Association, Berkeley, CA, USA, Article 1, 2007, pages. 14.
[26] í.S. Cunha, I. Viana, J. Palotti, J.M. Almeida, V. Almeida, Analyzing security and energy tradeoffs in autonomic capacity management, in: Proc. NOMS, 2008, pp. 302–309.



Jianxin Li is an associate professor in the School of Computer Science and Engineering, Beihang University, Beijing China. He received the Ph.D. Degree in Jan. 2008. He has authored over 30 papers in IEEE T. on Industry Electronic, Information Sciences, SRDS and HASE etc. His research interests include trust management, information security and cloud computing system. He is a member of IEEE.

Bo Li received his Bachelor and Master degrees from Dalian University of Technology, China. Now he is a Ph.D. student at the Department of Computer Science, Beihang University, China. His research interests include a broad range of topics related to computer security including virtualisation security, operating system security, and trusted computing.

Tianyu Wo received his B.Eng. and Ph.D. Degrees both in computer science from Beihang University, China, in 2001 and 2008 respectively. He's now an assistant professor of the School of Computer Science and Engineering, Beihang University. His current research interests include large scale distributed systems, virtual computing environments, network operation systems and network enabling applications. He is a member of IEEE.

Chunming Hu received his B.Eng. and Ph.D. Degrees both in computer science at Beihang University in 1999 and 2005. He is a research staff member and Associate Professor at the School of Computer Science and Engineering, Beihang University, Beijing, China. He's published more than 30 papers including the best paper award in ICEBE 2005. His research interests include peer-to-peer and grid computing; distributed systems, virtual computing and software architectures.

Jinpeng Huai is a Professor and President of Beihang University. Prof Huai is Academician of Chinese Academy of Science. He serves on the Steering Committee for Advanced Computing Technology Subject, the National High-Tech Program (863) as Chief Scientist. He is a member of the Consulting Committee of the Central Government Information Office, and Chairman of the Expert Committee in both the National e-Government Engineering Taskforce and the National e-Government Standard office. Prof. Huai and his colleagues are leading the key projects in e-Science at the National Science Foundation of China (NSFC) and Sino-UK. He has authored over 100 papers. His research interests include middleware, peer-to-peer (P2P), grid computing, trustworthiness and security.

Lu Liu is the Senior Lecturer at the School of Computing and Mathematics, University of Derby (UK). Before joining the University of Derby, he was a Lecturer at the School of Engineering and Information Sciences at Middlesex University (UK). Prior to his academic career, he was a Research Fellow at the School of Computing at the University of Leeds (UK), working on the NECTISE Project which was an UK EPSRC/BAE Systems funded research project involving ten UK Universities and the CoLaB Project which was funded by UK EPSRC and the Chinese 863 program. He received a Ph.D. Degree (funded by UK DIF DTC) from the University of Surrey (UK) and M.Sc. Degree from Brunel University (UK). His research interests are in the areas of service-oriented computing, software engineering, Grid computing and peer-to-peer computing. Dr. Liu has over 40 scientific publications in reputable journals, academic books and international conferences. He won the Best Paper Award at the Realising Network Enabled Capability Conference in 2008. He is member of IEEE.

K.P. Lam joined the Computer Science course at Keele University in 1997/98 and has extensive experience of working in industry. His research interests have a significant overlap in the fields of digital multimedia, computer vision/visual analytics and biometric security technologies for distributed systems. He was a major grant holder of the UK/EPSRC funded project, entitled Element Specific X-ray Imaging for Security applications (2005–07) jointly with the School of Chemistry and Physics, and Criminology, and latterly, the UWSP/AWM (UK) Proof of Concepts funded project that concerned the joint development of a digital media based authentication system with a West Midlands company. Currently, he is working on the EPSRC/Keele funded project (3ME) to develop innovative methods of visualising and analysing biomedical data/images including MRI scans, X-rays/FT-IR spectra, and live cells. He is also the Editor in Chief of the (new) Journal of Cloud Computing (JCC), IBIMA Publishing.