Nonlinear Phenomena in Complex Systems, vol. **17**, no. 3 (2014), pp. 253 - 262

Formal Security Model for Virtual Machine Hypervisors in Cloud Computing Systems

D. P. Zegzhda and A. V. Nikolsky
Saint-Petersburg State Polytechnic University,
29 Politechnicheskaya Str, 195251 Saint-Petersburg, RUSSIA

(Received 27 March, 2014)

The paper describes a formal security model for virtual machine hypervisors in cloud systems based on the graph theory. This model defines security properties and data access operations hierarchy inside virtual machines and hypervisors in the cloud. The proposed model makes it possible to formalize major security issues for cloud systems and define tasks for hypervisor security and methods to solve them as well as the security condition for the virtual machine hypervisors.

AMS Subject Classification: 68R10, 94C15
Keywords: cloud computing, hypervisor security, virtualisation, graph theory, virtual machine

1. Introduction

Cloud computing is a widely implemented and developing business model that provides users with the access to distributed computing and information resources as a single service. However, access to this service can be provided via the Internet, in the case of public clouds, and via a local network of the organization, in the case of private clouds. In both cases, a user of the cloud service processes data using temporarily provided computing resources but can not control the system architecture in which these resources are located. These are the main advantages and risks of cloud computing. When using cloud computing, user data are transferred to the "cloud" and the control over the data is lost as the provider does not bear responsibility for its safety [1]. This significantly complicates the development and implementation of security systems for cloud computing, therefore the revision and adaptation of current security models is required. Security of a large number of hypervisors running in the cloud affects significantly the safety of the entire cloud computing system [1]. It is the hypervisor that controls operations of several virtual machines, therefore it is to provide the isolation of virtual machines from each other and from internal network components of the cloud. The main features of the hypervisor virtual machines in cloud computing systems are as follows:

- use of virtual machines as hosts in the cloud;
- migration of virtual machines;
- replication of virtual machines;
- storage of virtual machine image files in a centralized repository;
- ability to connect virtual machines into isolated virtual networks.

Current models do not always describe these features of virtual machines operation in the cloud, so they are not applicable for formalization of security requirements for cloud computing systems. The purpose of this paper is to develop a formal security model for cloud systems, that will take into account the features of hypervisors and virtual machines application in the cloud and identify conditions for hypervisor security in the clouds.

2. Review of current models for cloud computing

Due to rapid development of cloud computing systems there is a growing number of mathematical models describing various features of such systems. Joe Vienman developed an axiomatic theory and worked out the cloud computing model based on it [2]. In his work he put forward one of the most generalized models for cloud computing. The model proposed by Vienman is universal and can be applied not only in the field of information technology, but also in several other fields (rental cars, electric networks, logistics and others), which gives it a major advantage. In this model the cloud is defined as a structure that meets the following requirements: it should be 1) general regardless of the resource type; 2) independent of location; 3) accessible at any time; 4) cost-effective; 5) available on request. An important role in the cloud is given to a time variant directed graph. Nodes of the graph represent sources and resource consumers in the cloud, and arcs define directed flows of these resources redistribution. Besides the graph theory, Vienman cloud model uses elements from other theories such as Markov processes and the theory of automated machines. In terms of the goal set by the authors, the absence of explicit elements of security systems, such as access subjects and objects, appears to be the main disadvantage of this model. V.Chan and his colleagues in their paper [3] propose a model based on the graph theory that enables to describe the behavior of programs and applications running in the cloud. The paper focuses on the issues related to testing applications in the cloud. The vertices of the graph in this model represent computing nodes characterized by a variety of attributes, and arcs in a graph serve as communication links between computing nodes. The specific feature of this model is its ability to connect the predicate to any arc of the graph, which enables to specify the condition of the resource use in the cloud. The model does not consider a virtual machine explicitly, therefore it is not relevant in terms of the goal set by the authors. Yuingmin Lee and Omar Boukalma put forward an approach for diagnosing sources of information flows in the cloud, based on the system simulation [4]. The paper focuses on security and verification of cloud computing systems. The proposed model is based on the Petri nets and allows for secure communication protocols between various system components. The model takes into account not only security features of links between nodes, but also the safety of the entire cloud system. However, the features of hypervisors and virtual machines in cloud computing systems are beyond the consideration of Lee and Boukalma model. Huai together with his colleagues in their work [5] complement the existing model of abstract state services (AS2s) by formalizing the service schedule. The application of this extended model makes it possible to analyze real sequence of operations explicitly carried out by the service. The article proposes service schedules in the form of regular expressions based on Kleene algebra. However, this paper discusses only cloud services leaving out various options for networking in the cloud. Thomas and his colleagues developed not only a model but also a system to simulate cloud networks [6]. Using the developed platform and software for its simulation, they confirmed the accuracy of their hypothesis: the cloud computing capacity will be used more efficiently if the service provider and a user are given the opportunity to select or change scheduling and computation distribution algorithms within the user resources, in addition to usual cloud services. The proposed model deals with the cloud system as a graph that consists of computing resources and communication links between them. Each resource and each link are specified by speed characteristics. The formalization of the computational concepts enabled the authors to check effectiveness of different ways of computing resources distribution in the cloud. However, this approach does not enable to analyze security of the system as it leaves out the concepts of subjects and objects in the model. Thus, graph theory enables to describe in detail a complex cloud structure with lots of network hosts and computing resources. However, the proper analysis of cloud systems safety requires a model that takes into account explicit features of hypervisors and virtual machines.

3. Formal security model of cloud computing

The proposed formal model of cloud computing system security focuses on hypervisors, virtual machines and their relationship with each other and other system components. Virtual machines and virtualization technology play a key role in cloud computing systems. Software platforms (e.g., VMware vSphere, Xen Cloud platform), based on cloud systems, define procedures and rules for interaction between hosts inside the cloud network [4]. Modern data processing centers based on cloud computing systems, can comprise up to thousands of hosts inside the cloud network. Each host performs a certain function in the cloud. Among the functions applicable for network hosts in the cloud, there are several major functions typical for all cloud computing systems. User data are located on multiple hosts inside the cloud network, called a storage. Users in the cloud get indirect access to their data through intermediaries - the virtual machines regardless of the way the service is provided (IaaS, PaaS or SaaS). Some of the hosts in the cloud internal network carry out functions of control systems: accounting and allocation of resources, development and control of virtual machines, centralized control of the network hosts inside the cloud. Most often, there is only one host responsible for all these functions, that is the controller. Most of the entire cloud internal network is taken by the worker hosts whose main function is to provide the operation of several virtual machines carrying out calculations requested by users. Thus, we can define a set R of all possible functions for the cloud internal network. This set contains the following elements:

- *VmHost* is a host with the hypervisor providing its hardware resources to multiple virtual machines;
- *Storage* is a host representing the database and storing user data as well as virtual machine images, virtual machine templates or program images;
- *Controller* is a control center responsible for supervising, monitoring and performing other functions necessary for the operation of the cloud internal network.

$$\{VmJost, Storage, Controller\} \subseteq R$$

As the network inside the cloud is actually a standard computer network, it can be represented by an undirected graph $C_L = (N, \eta)$, where $N = \{n_i\}$ is a set of hosts inside the cloud network, $\eta \subseteq N \times N$ is a set of facets of the graph indicating the authorized network connections between network hosts. η is determined by the physical network connections as well as settings of firewalls running in the cloud internal network. So, the function *Role* assigns a set of roles to each host: $Role : N \to P(R)$. In large data processing centers each host performs only one role ($\forall n \in N, |Role(n)| = 1$), though the software for cloud computing systems enables to assign several different roles to one host (e.g. the role of the control center and the storage at the same host). Let us denote the set of all virtual machines in the cloud as follows: $V = \{\nu_i\}$. Similarly to the cloud internal network graph we can build the network graph formed by the virtual machines in the cloud: $C_V = (V_\upsilon)$, where $\upsilon \subseteq V \times V$ is a set of authorized virtual network connections between virtual machines. υ is determined by the settings of virtual networks in the cloud. Any element of N is the hardware for running software in cloud computing system (including operating systems, hypervisors and database management systems). This software implements a network protocol stack (generally, TCP/IP), thereby forming an independent host inside the cloud network. Each virtual machine on the *VmHost* host implements its own network protocol stack making virtual machines independent of network hosts. Connecting virtual machines with the cloud internal network hosts from set N, we introduce the concept of "machine". We denote the set of all machines in the cloud as $M \subseteq V \cup N = \{m_i\}$. So, the full graph of the cloud network can be determined as follows: $C = (M, c)$, where $c \subseteq M \times M$ and $c \subseteq \eta \cup \upsilon$. Thus, the model comprises three graphs: C_L, C_V and C. C_L, being a connected graph, represents the hardware resources of the cloud and the relationships between them. This graph is a core one and enables the cloud computing system to provide users with an access to various services through the external network. IaaS cloud service users interact with the sets of virtual machines connected into isolated virtual networks. Consequently, the graph C_V will not be connected in this case. It should be noted that graph C_V will be isolated from graph C_L ($\forall \nu_j \in V : \neg \exists n_i \subset N : (\nu_j, n_i) \in c$) in case there is no networking between virtual machines and the cloud internal network, which is typical for public clouds.

3.1. Virtual machines in the cloud

Apart from networking in column C, there are other connections formed through the use of virtualization technology between the set of virtual machines V and N-cloud internal network hosts. Any virtual machine can be represented as a regular file (or a set of files), i.e. a virtual machine image. The image file of the virtual machine can store its copy of RAM as well as hard disk image and additional settings. Each virtual machine in the cloud has its own image file located in the storage. The virtual machine image file can be distributed across multiple hosts in the network instead of being stored on one host. If the virtual machine is running, it runs on one of the worker hosts. If the virtual machine is not in operation, it is still present in the cloud as an image file. A virtual machine can have the replication function, which allows for rapid recovery of virtual machines in case of a host failure.Let us formalize these relationships as follows:

- $H \subseteq V \times N$ is a set of pairs (ν_i, n_i), where ν_i is a virtual machine running on a node h_i inside the cloud network. This set defines all virtual machines currently running in the cloud and corresponds to the following: $VmHost \in Role(h_i)$;

- $I \subseteq V \times P(N)$ is a set of pairs (ν_i, S_i), where ν_i is a virtual machine and S_i is a set of all nodes in the cloud network to store the image file of the virtual machine with the condition $\forall s_i \in S_i, Storage \in Role(s_i)$;

- $F \subseteq V \times P(O)$ is a set of pairs (ν_i, O_{ν_i}), where ν_i is a virtual machine and O_{ν_i} is a set of files (from the set of all objects in the system O), which represent the image of the virtual machine in the cloud. All these files can be distributed over multiple hosts of the set I in the network;

- $R \subseteq V \times P(N)$ is a set of pairs (ν_i, T_i), where ν_i is a virtual machine and T_i is a set of hosts inside the cloud network with the replication function for this virtual machine, which follows the condition: $\forall t_i \in T_i, VmHost \in Role(t_i)$.

Let us denote the set of all programs in the cloud as $P = \{p_i\}$. Software running on host m_i may be represented as $P_{m_i} = \{p_j^{m_i}\}$, with the following condition: $P \subseteq \bigcup P_{m_i}$. Assuming that $Soft \subseteq M \times P(P)$ is a set of connections between the machines in the cloud and the programs running on them we can introduce two subsets in this set:

- $Soft^N \subseteq N \times P(P) \subset Soft$ is a set of programs running on the hosts inside the cloud network;

- $Soft^V \subseteq V \times P(P) \subset Soft$ is a set of programs running in virtual machines in the cloud.

Virtual machine is an abstraction created by the hypervisor software, which is a program that runs on the host hypervisor:

$$\forall(\nu_i, h_j) \in H, \nu_i \in V, h_j \in N :$$
$$\exists P_{n_j} \subset P, (\eta_j, P_{n_j}) \subseteq Soft^N \wedge \nu_i \in P_{n_j}$$

It can be stated that $V \subseteq P$. It should be noted that a set of programs P contains not only the programs currently running in the cloud, but also images of the programs stored in the template files of virtual machines that are used to create new virtual machines in the cloud. Therefore, if the virtual machine ν_i is not running on any host in the network cloud ($\nexists h_j \in N : (\nu_i, h_j) \in H$), then it is still the program ($\nu_i \in P$), but in the form of its image.

3.2. Objects and users in the cloud

Before formalizing the state of a cloud computing system it is necessary to introduce safety-related system definitions. We can use definitions from [7] adapting them to the cloud systems:

- $U = \{u_i\}$ is a set of users in the cloud;
- $S = \{s_i\}$ is a set of objects in the cloud. The set of objects in the cloud is determined by the software P;
- $Id \subseteq U \times P(S)$ is an identification set. This is a set of of pairs, assigning each user in the cloud to the set of all their subjects;
- $Imp \subseteq P \times S$ is an impersonation set. It assigns each program running in the cloud to the only entity that determines its rights in the system.

Using these sets in the proposed cloud model makes it possible to take into account security issues within the cloud.

3.3. The security model of cloud computing

The state of the cloud computing system can be represented as a tuple: $\psi = (C, P, U, S, H, I, R, Soft, Id, Im)$, where:

- C is a graph representing all network connections between all hosts inside the cloud network and virtual machines;
- P is a set of all programs in the cloud and virtual machines;
- U is a set of all users of the cloud;
- S is a set of all objects in the cloud;
- $H, I, R, Soft, Id, Im$ are sets of tuples that define the current relationship between users, programs, objects and network hosts in the cloud.

With these definitions we can represent any cloud computing system in the form of a finite state machine $\Omega = (\Psi, \psi_0, \tau)$, where Ψ is a set of all system states with $\psi_0 \in \Psi$ being the initial state of the system, and $\tau : \Psi \to \Psi$ is a function of the system transition from one state to another. One of the main features of the cloud provided by the virtualization technology is the migration of virtual machines between different network hosts in the cloud during their operation. This virtualization function is referred to as live migration [8]. The virtual machine migration causes set H to change with time. Migration can be represented in the proposed model as an operation of the cloud computing system state changes (from state ψ to state $\widetilde{\psi}$), with the following assignments:
$H^{\widetilde{\psi}}{=}H^{\psi}\backslash(\nu_i, n_j){\cup}(\nu_i, n_k), \nu_i{\in}V^{\psi}, n_j{\in}N^{\psi}, n_k{\in}N^{\psi}$
$\psi{=}(C^{\psi}, P^{\psi}, U^{\psi}, S^{\psi}, H^{\psi}, I^{\psi}, R^{\psi}, Soft^{\psi}, Id^{\psi}, Im^{\psi})$
$\widetilde{\psi}{=}(C^{\widetilde{\psi}}, P^{\widetilde{\psi}}, U^{\widetilde{\psi}}, S^{\widetilde{\psi}}, H^{\widetilde{\psi}}, I^{\widetilde{\psi}}, R^{\widetilde{\psi}}, Soft^{\widetilde{\psi}}, Id^{\widetilde{\psi}}, Im^{\widetilde{\psi}})$
Thus, the proposed model allows us to describe the basic features of hypervisors and virtual machines in the cloud such as:

- use of virtual machines as hosts in the cloud is expressed as $V \subseteq M$;

- migration of virtual machines is expressed in the model as $H^{\widetilde{\psi}}{=}H^{\psi}\backslash(\nu_i, n_j){\cup}(\nu_i, n_k), \nu_i{\in}V^{\psi}, n_j{\in}N^{\psi}$, $n_k{\in}N^{\psi}$;
- replication of virtual machines is expressed in the model as $R \subseteq V \times P(N)$;
- distribution of the storage environment and performance of virtual machines is described as: $\forall(\nu_i, S_i){\in}I : \nexists n_i{\subset}N : VmHost{\in}Role(n_j)$;
- the capability to connect virtual machines into isolated virtual networks is provided with a separate disconnected graph C_ν.

Connecting cloud computing network and virtual machines in a single graph makes it possible to use the mathematical apparatus of the theory of graphs for the more detailed simulation of cloud systems. Moreover, with the proposed model, we can work out the rules and security policies that impose restrictions on the virtual machines and programs and the hosts inside the cloud network. Thus, the requirement for the absence of networking between the virtual machines in the cloud and the hosts inside the cloud can be described as follows: $\forall \nu_j \in V : \nexists\ n_i \subset N : (\nu_j, n_i) \in c$.

4. Hypervisor security in the cloud

As it was mentioned above, any virtual machine is a program running under the control of a hypervisor with each virtual machine running the programs. These statements can be written as follows:

$$\begin{cases} V \subseteq P; \\ \forall \nu_j \in V : \exists P_{\nu_j} \subset P, (\nu_j, P_{\nu_j}). \end{cases}$$

These statements imply:

$$\forall(\nu_i, n_j) \in H, \nu_i{\in}V, n_j \in N, (\nu_i, P_{\nu_i}) \in Soft, p_{\nu_i} \in P_{\nu_i} :$$
$$\exists(u, S_u){\in}Im, u{\in}U, S_u{\subset}S, \{s_\nu, s_n\}{\subset}S_u, (\nu_i, s_n){\in}Id \wedge (p_{\nu_i}, s_\nu){\in}Id.$$

This means that any user u in the cloud is assigned to at least two different subjects:

- s_ν is a subject that is assigned to program p_{ν_i} running in the virtual machine ν_i. Subject s_ν is involved in the security system inside the virtual machine ν_i (security in the operating system);
- s_n is a subject that is assigned to program ν_i running on host n_j. Subject s_n is involved in security system running on host n_j (hypervisor security).

It should be noted that access rights of these two subjects are controlled by two different security systems. Access to the resource is provided on behalf of the subject s_n. Hypervisor is responsible for the operations performed by the software within the virtual machine using a variety of other operations performed by the software on host n_j.

4.1. The hierarchy of data access operations from the virtual machine in the cloud computing systems

Hard disks of virtual machines in cloud computing systems are often stored on separate servers in the cloud, rather than on the hypervisor host, with the access to hard disks provided only via network [9]. This means that for each access to a virtualized disk from a virtual machine you must perform a number of operations to access a file on a remote server from the internal cloud network. The virtualized disk is represented in the cloud computing system as an integral part of a virtual machine image stored as a file or a set of files in the central repository inside the cloud network. Consequently, the direct access to data will be carried out by the software running on the central repository. Let us consider this in more detail and introduce some definitions: ν_i is a virtual machine that provides access to the file; f is the file located on the virtual hard disk which is being accessed; p_{ν_i} is a program that provides access from a virtual machine; n_r is a centralized repository storing the virtual machine image. We denote data access operation performed by the program p_{ν_i} as op^{ν}. In order to perform this operation the program p_{ν_i} accesses the operating system running inside the virtual machine ν_i. A security subsystem in the operating system first gets access control for this operation using subject s_ν (which corresponds to program p_{ν_i}). If the operation is enabled, the operating system performs a set of low-level operations $\{op_i^{\nu}\}$ to copy the data to a virtual device (the virtual hard disk with file f). Low-level operations including the interaction with hardware IO ports and DMA transaction processing are performed by the file system drivers. All these operations are carried out at the privileged level inside the virtual machine and do not use object f. Instead, hardware I/O ports and memory areas are the objects of these operations. Hypervisor controlling a virtual machine ν_i performs a set of data access operations $\{op_i^h\}$ implemented in the storage device emulators (on the virtual hard disk) for all operations from the set $\{op_i^{\nu}\}$. It should be noted that all operations from set $\{op_i^h\}$ are performed directly on the host inside the cloud network as they need to get access to the real (non-virtualized) hardware. Therefore, all operations from the set $\{op_i^h\}$ are performed using subject s_n (corresponding to the program of a virtual machine ν_i) and go through the control system in the hypervisor. The composition of the set of operations $\{op_i^h\}$ depends on the settings and the specific implementation of the hypervisor. Since the virtual hard disk image of the virtual machine is stored in the repository, the following conditions are fulfilled: $\exists\ T_\nu : n_r \in T_\nu \wedge (\nu_i, T_\nu) \in I$ and $\exists W_\nu : \{f_i^{\nu}\} \subseteq W_\nu \wedge (\nu_i, W_\nu) \in F$, where n_r is the host storing the image of the virtual machine hard disk (represented as a set of files $\{f_i^{\nu}\} \subseteq O$). Then, there is a subject s_r on whose behalf the access to the virtual machine image files is provided on the remote server n_r. Thus, the operations $\{op_i^h\}$are responsible for data exchange over the network, using the link $(n_i, n_r) \in \eta$ in the graph C_L. These actions result in the set of operations $\{op_i^{\gamma}\}$ for data access of the image of the virtual machine ν_i performed on the host. The given example shows that the user in the cloud computing system may be assigned to at least three different subjects $\{s_\nu, s_n, s_r\}$, which correspond to three different programs $\{p_\nu, \nu_i, p_r\}$ (p_r is a program running on host n_r and providing access to a virtual machine image data), which, in their turn, are running in three different environments $\{\nu_i, n_i, n_r\}$. Low-level operations from the set $\{op_i^{\nu}\}$ are not controlled by any security system, instead they generate two sets of operations: $\{op_i^h\}$ and $\{op_i^{\gamma}\}$. Thus, we can build a hierarchy of IO operations in the cloud computing system to access the file on the disk from the virtual machine (Table 1). So, when the virtualization technology is applied in cloud computing systems the hierarchy of access operations takes on a specific feature: all operations performed in the hypervisor by the virtual machine device emulators are authorized with the user's rights of the virtual machine program in the hypervisor, but not with the user's rights of the program within the virtual machine.

4.2. Hypervisor security issues in the cloud computing systems

One data access operation within a virtual machine generates a sequence of other data access operations in the hypervisor on a remote server, which changes the subject, object and the type of the access. The interaction of the client and server programs in the network can

Table 1: Hierarchy of access operations to the disk file from a virtual machine in the cloud

Operating environment	Program	Operations	Object	Object identifer	Subject
ν_i-virtual machine	p_ν	op^ν	f-file on the virtual disk	file name on the virtual disk	s_ν
ν_i-virtual machine	kernel of the operating system in a virtual machine	$\{op_i^\nu\}$	disk sectors or memory areas	file corresponding sector numbers on a virtual disk	operating system kernel
n_i-worker host	ν_i	$\{op_i^h\}$	$\widetilde{f}$-network socket	socket number corresponding to the server with storage	s_n
n_i-worker host	hypervisor on a worker host	operations of interaction with the network card device on a worker host	memory areas and a network interface	network address of the server with storage	hypervisor
n_r-server storage of virtual machine images	p_r	$\{op_i^\gamma\}$	$\{f_i^\nu\}$-virtual disk	image filename of a virtual machine hard disk	s_r
n_r-server storage of virtual machine images	operating system kernel on the repository	operations of interaction with the information storage device on the repository	disk sectors and memory areas	sector numbers on the virtual disk corresponding to the sector numbers on the hard disk	operating system kernel

cause a similar situation. In the process of interaction over the network the subject on the client side and the server side belong to the same user as the identification and authentication network mechanisms are used. In the case of virtualization the operating system in a virtual machine monitors the implementation of data access operations of different users at the level of the processes inside, and the hypervisor host system does not control these operations because it provides the virtual machine operation as a single program running for one user. This can result in two system states that can be used to carry out attacks on the virtualization system [10]. In the first case, if the cloud computing system is used by the SaaS users, it can provoke the situation in which two users work in the same virtual machine, while the direct data access operations will be performed on behalf of one subject (the one assigned to the virtual machine which is running on a worker host). Consequently, many different users in a virtual machine may be assigned to the subject on the host system, which does not correspond to any of these users:

$$\exists (u1, S_{u1}) \in Im, (u2, S_{u2}) \in Im, u1, u2 \in U, S_{u1},$$
$$S_{u2} \subset S, s_{\nu 1} \in S_{u1}, s_{\nu 2} \in S_{u2},$$
$$\{(p_{\nu 1}, s_{\nu 1}), (p_{\nu 2}, s_{\nu 2})\} \subset Id, (\nu_i, P_{\nu_i}) \in Soft,$$
$$\{p_{\nu 1}, p_{\nu 2}\} \in P_{\nu_i}, (\nu_i, s_n) \in Id, s_n \notin S_{u1} \wedge s_n \notin S_{u2}.$$

The second problem can arise if the set of virtualization facilities such as Xen or KVM, implements abstraction of the virtual machine using multiple components running on behalf of different subjects. For example, in KVM any virtual machine can run in the user mode, but, for a number of operations performed by the software in a virtual machine, a special emulator process qemu, running with the administrator rights of the host system, is used [11]. Thus, part of the processes initiated by the user in a virtual machine will be run on behalf of the administrator of the host system. Consequently, the non-privileged user in a virtual machine can get more privileges in the host system [12]. In the framework of the proposed model this situation can be expressed as follows:

$$\exists (u1, S_{u1}) \in Im, u1 \in U, S_{u1} \subset S, s_{\nu 1} \in S_{u1},\\ (p_{\nu 1}, s_{\nu 1}) \in Id, (\nu_i, P_{\nu_i}) \in Soft,\\ (\nu_i, s_n) \in Id, s_n \in S_{u1}.$$

Summarizing, we can assert that if the subject inside a virtual machine has the access to virtual resources related to the image of the virtual machine, the access to these resources will be controlled only at the level of the operating system inside the virtual machine, and the access operations will be carried out in another runtime without the access control. In order to ensure the safety of the hypervisor and virtual machines in cloud computing we need to solve the following problems:

- authorization and access control of all user requests to the cloud system resources from within the virtual machine;

- minimization of the privileges of device emulators that provide the access of the virtual machine to the resources in the cloud or hypervisor.

To solve the first problem we suggest using network protocols with the authentication of access to cloud resources and placing these resources outside the virtual machine. In this case, programs processing SaaS queries and running inside the virtual machine will not be able to modify or read any local data of the virtual machine they operate in, but they will be able to perform queries to an FTP server or the database running inside the cloud. To solve the second problem we propose to isolate all device emulators that provide access to the cloud resources (e.g., hard disk emulators) within the processes running with fewer privileges. In the case of attack at device emulators thereby isolated the violator can not affect the operation of the hypervisor or access the internal cloud network. So, we can formulate the security condition for cloud computing systems: device emulators that provide access to the cloud resources must be enabled with the privileges of the related virtual machine user, and all requests to the cloud resources from the virtual

5. Conclusions

The proposed formal model of the virtual machine hypervisor security in cloud computing systems enables to describe cloud systems entities and relationships between them as well as the process of the cloud functioning in terms of security. It describes the security of the hypervisor basic functions and the operations with virtual machine, such as migration and replication of virtual machines, placement of virtual machines images on the repository and representation of the virtual machine as a program and as a network host. The analysis of the hierarchy of data access operations from the virtual machine in the cloud computing system enabled to identify and describe two main hypervisor security flaws in cloud computing systems in terms of the proposed model: the inability to identify requests for access to the cloud resources outside the virtual machine, and the elevated privileges in the hypervisor when processing I/O requests from the virtual machine. Consequently, in the view of the need to eliminate these deficiencies, we formulated hypervisor security issues in cloud computing systems, proposed the solutions, and formulated the condition of the system security for cloud computing, the observance of which will ensure the invariance of the user's authority within the virtual machine and beyond the cloud environment. The proposed model can be used as a formal basis for setting the criteria of security policies for cloud computing systems, restrictions on migration and replication functions as well as on the access to the software and virtual machines.

References

[1] P.Zegzhda, D.Zegzhda, A.Karetnikov. Cloud computing. Virtual safety or virtualization? In: *Proceedings of the Conference "RusCrypto-2012* , Moscow 28–31 march 2012. [Electronic resource] http://www.ruscrypto.ru/sources/conference/rc2012/

[2] D.Vienman. Axiomatic theory of clouds. [electronic resource] Slave. Doc, july 2011. http://www.joeweinman.com/Resources/Joe_Weinman_Axiomatic_Cloud_Theory.pdf

[3] W.K. Chan, L. Mei, Z. Zhang. *Modeling and testing of cloud applications.* (IEEE APSCC, Singapore, 2009). Also at http://www.cs.cityu.edu.hk/∼wkchan/papers/apscc2009-chan+mei+zhang

[4] I. Lee, O. Bukelma. Model of CPN origin workflow: Towards diagnosis in the cloud [electronic resource]. http://ceur-ws.org/Vol-789/paper6.pdf

[5] H. Ma, K. Scheve, Talheym B., C. Wang. *A formal model of collaboration in serving the clouds.* SOCA, vol. 6, no. 3. (Springer, 2012). Also at http://cdcc.faw.jku.at/pdf/get.php?f=FM_Cloud.pdf&d=/public/publications

[6] T. Henzinger *et al.* FlexPRICE: flexible resourcing with the cloud [electronic resource]. http://pub.ist.ac.at/∼wies/papers/flexprice.pdf

[7] P.Zegzhda, D.Zegzhda. Methodology of dynamic protection. In: *Proceedings of the International Conference on Security and Counter Terrorism.* (MTNMO, Moscow, 2006).

[8] K. Clarke *et al. Live migration of virtual machines.* Eds. K. Clark and [etc.] (University of Cambridge, 2005). Also at http://www.cl.cam.ac.uk/research/srg/netos/papers/2005-nsdi-migration.pdf

[9] M.T. Jones. *Anatomy of cloud storage infrastructure.* (IBM, 2010). Also at http://www.ibm.com/developerworks/cloud/library/cl-cloudstorage/cl-cloudstorage-pdf.pdf

[10] P. Zegzhda, D. Zegzhda, A. Karetnikov. Security of cloud computing systems. Problems and prospects. In: *INFOFORUM, 7 february 2012.* (MiFi Publishing House, Moscow, 2011). P. 116-118.

[11] D.P. Berrange. Taking full advantage of QEMU userspace Xen. Conf. Xen Summit, november 2007. [electronic resource]. http://people.redhat.com/berrange/xen-summit-2007-sj/xen-summit-xenite-report.pdf; http://people.redhat.com/berrange/xen-summit-2007-sj/xen-summit-xenite-report.pdf

[12] N. Elhage. Virtunoid: Guest mode KVM. Escalation of existing committers to host. Conf. Black Hat, 2011. [electronic resource] http://media.blackhat.com/bh-us-11/Elhage/BH_US_11_Elhage_Virtunoid_WP.pdf

[13] Cloud computing. Merits, risks and recommendations for information security. Eds. D. Katteddu, G.Hokben; (ENISA, 2009). Also at http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment/at_download/fullReport