ICIC Express Letters
Volume **14**, Number **4**, April **2020**

 ISSN 1881-803X
pp. **387–397**

A LOGGING SYSTEM IN OPENSTACK ENVIRONMENT TO MITIGATE RISKS ASSOCIATED WITH THREATS IN INFRASTRUCTURE AS A SERVICE CLOUD

Wichep Jaiboon[1], Winai Wongthai[1,2,*], Thanathorn Phoka[1] and Thongrob Auxsorn[3]

[1]Department of Computer Science and Information Technology
[2]Research Center for Academic Excellence in Nonlinear Analysis and Optimization
Faculty of Science
Naresuan University
99 Moo 9, Tambon Tha Pho, Muang, Phitsanulok 65000, Thailand
*Corresponding author: winaiw@nu.ac.th

[3]Department of Information Technology
Faculty of Science and Technology
Pibulsongkram Rajabhat University
156 Mu 5 Plaichumpol Sub-district, Muang District, Phitsanulok 65000, Thailand

Received October 2019; accepted January 2020

Abstract. *An Infrastructure as a Service or IaaS cloud is rentable virtual computing resources such as networking, servers, and storage that are offered by a cloud provider to its customers. The customers can use these resources through the Internet. However, the Cloud Security Alliance (CSA) indicates that there are security concerns of an IaaS such as confidentiality, integrity, and availability of customers' files in the IaaS. These concerns can affect both providers and customers. Currently, open-source software for creating IaaS clouds such as OpenStack has increasingly been applied in the production. However, an IaaS cloud that is built from software also yields the same security concerns as above. Thus, for the aim to mitigate the risks associated with the most serious CSA threat, this paper applied our previous logging system into OpenStack environment. This threat is a data breach which can be an incident when IaaS customer's sensitive or critical files are unauthorizedly opened, stolen, or used by unauthorized users. These files may contain sensitive data, such as health information. The results of this paper are that we delivered a design and implantation of a logging system architecture in OpenStack. Then, we illustrated the results of the delivered system and the discussions of the results. In real-world IaaS cloud production, the log files that were produced from the proposed logging system can be one of the solutions to assist the providers and customers in analyzing the risks above. Thus, the proposed logging system in the OpenStack environment for an IaaS cloud can help to mitigate risks associated with this CSA threat. This increases the reliability of an IaaS cloud to benefit both the providers and customers.*

Keywords: Cloud security, Logging system, OpenStack, Monitoring system, IaaS, Hypervisor

1. **Introduction.** National Institute of Standards and Technology [1] states that cloud computing or cloud is a model for using computing resources (such as networks, servers, storage, and applications). It also argues that these resources can be conveniently accessed from a variety of devices and connected via the network from anywhere. The provision and cancellation of these resources can be done quickly and adjusted according to needs, with minimum management effort or contact with the service provider [1]. The cloud is increasingly used in the organization because it can be scalability and elasticity as argued in [2]. The cloud can reduce the cost of information technology management in

DOI: 10.24507/icicel.14.04.387

an organization as discussed in [3, 4, 5, 6, 7, 8, 9, 10]. We focus on a public cloud. This cloud is provided for open use by the general public or cloud customers, can be operated, managed, and owned by business companies or cloud providers [1]. In this paper, the word 'a cloud' refers to 'a public cloud'. Infrastructure as a Service or IaaS cloud is provided by a provider for customers processing, storage, networks, applications, operating systems and other fundamental computing resources which the customers can adjust according to their needs [1]. This paper refers to 'an IaaS cloud' as only 'an IaaS'. An IaaS is increasingly used in many applications domains. Gartner predicts that in 2021, an IaaS market growth can be $83.5 billion [11].

However, there are still security concerns when using the IaaS cloud on issues of confidentiality, integrity, and availability of customer's sensitive files as argued by [8, 12]. The CSA published the treacherous 12: cloud computing top threats in 2016 report [13] that identifies the security concerns above. The concerns from the report can affect both the IaaS cloud providers and customers. The customers want to know where their files are stored or who has accessed the files. The provider should also provide the log data or evidence as proof for the customers when they need in case there is something wrong with the files [8]. To build an IaaS cloud, a provider can use a cloud software platform such as OpenStack to build this cloud with both copyright and open-source. This paper focuses on OpenStack, which is also the increasing trend of using open-source software in the business organization, as argued in [14]. From the security concerns mentioned above, we proposed in our previous work [8] a logging system to mitigate risks associated with CSA threats for IaaS cloud.

Research gaps: Firstly, rather experiment in OpenStack, logging systems from our previous work [8] were experimented in a simulated cloud environment in a single physical computer machine. These logging systems worked well; however, they need to work in a real-world production environment such as in OpenStack. Thus, in this paper, we aim to enable our logging systems from our previous work to perform in an OpenStack environment with the primary objective of mitigating the risks associated with a CSA threat for IaaS cloud. Secondly and thus, from the first research gap, there is no design and implementation of a logging system in OpenStack in the literature. We provide both the design and implantation. Thirdly and then, there are no results and discussions to illustrate how the results from design and implementation could help in mitigating the risks above. With an IaaS perspective, thus our research question is how can we enable our previous logging systems to work in an OpenStack environment.

Summary of contributions: This paper has the following four contributions. The first, in Section 2, we discuss how an IaaS cloud and our previous logging systems can be compatible with OpenStack environment. This is to ensure that our previous systems could be applied to the new OpenStack environment. Secondly, based on the first contribution, in Section 3, we apply our previous logging system into the new OpenStack environment. This application includes the design and implementation of a logging system in the OpenStack environment for an IaaS cloud. This ensures the technical details of the experiment to apply our previous works to the OpenStack environment. The third, Section 4 illustrates the results from the design and implementation from Section 3. The results confirm that our previous logging systems can be applied to the OpenStack environment or an IaaS OpenStack cloud. Finally, the fourth, Section 5 discusses the design and implementation from Section 3 and the results from Section 4. The examples of the discussions are why our logging system can still work in a real-world IaaS cloud which is built from OpenStack or an IaaS OpenStack cloud, and our logging system in an IaaS OpenStack cloud can be alternative solutions apart from OpenStack security features to mitigate risks associated with the CSA top severe threat. Interestingly, we found other interesting discussions such as our logger can work with many hypervisors. All the contributions may help to mitigate risks associated with the first CSA threat or data breathes

in real-world IaaS cloud production, and maybe in other types of cloud computing such as PaaS and SaaS. Thus, to truly enhance security in a real-world IaaS cloud, this paper can be a guideline for mitigating first CSA threat and other eleven CSA threats.

2. Background.

2.1. Infrastructure as a Service architecture.

Figure 1 shows the IaaS architecture, which consists of two parties: the provider side and the customer side. The provider side is an organization (such as Amazon Elastic Compute Cloud) that offers various rentable services such as virtual machines or VMs to customers. The customer side is an organization or person that can access the services via the Internet. The main components of an IaaS architecture are hw, hypervisor, dom0, and domU. Hw is a physical computer machine to host all the rentable VM services, see the box at the bottom of the provider side of Figure 1. It is owned, managed, maintained by the provider. A hypervisor is a software that can enable the hw to run multiple VMs at the same time and in one hw, see the box in the middle on the provider side of Figure 1. See the box in the top left corner on the provider side of Figure 1, a dom0 stands for domain 0 and is a manager of all the VMs or domUs. It is an exclusive privilege domain of hypervisor. This means that it can access directly to the hw and can manage all the domUs. This domain will be started at the system boot. A domU or user domain, see domU1 and domUn in Figure 1, is an un-privilege domain for the customers to rent. A domU is running over the hypervisor and cannot access directly to the hw.

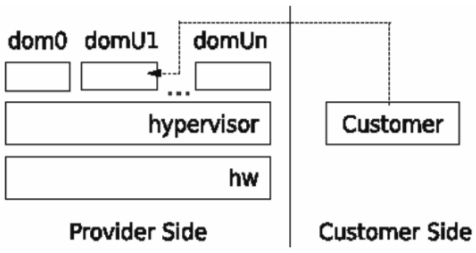

FIGURE 1. IaaS architecture from [8]

2.2. An OpenStack architecture.

From Figure 1, we will focus on the provider side by using OpenStack to create an IaaS public cloud to be close to the real-world production environment, in Section 2.3 and 3. [15] has compared the software used to create an IaaS public cloud such as Eucalyptus, OpenNebula, and OpenStack, using conditions including popularity, community, modularity, openness, and open-source. The results of this comparison show that OpenStack is the most appropriate option with scalable, compatible and flexible, and open-source features. OpenStack is for service providers, government agencies, enterprises, and academic institutions that want to build a cloud. This paper discusses OpenStack based on an IaaS provider's perspective. [16] states that OpenStack is a cloud operating system or OS that controls large pools of storage, networking, and compute resources in every part of a data center. The resources are managed via a dashboard that can enable the cloud provider's administrators to control while allowing their users to the resources via a web interface [16]. OpenStack is also open-source software that can be used to build an IaaS public cloud, and it can be used in the real-world cloud production environment. This cloud software has an increasing use in large business organizations [14]. Thus, we act as an IaaS provider then use OpenStack to create an IaaS public cloud to be close to a real-world cloud production environment. Then, we can and will apply our existing logging systems to this environment for our experiment in Section 3.1. See five boxes in Figure 2, there are five services of the architecture of OpenStack for an IaaS cloud [15], for short an IaaS OpenStack. These

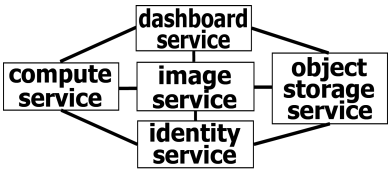

FIGURE 2. OpenStack architecture (from [15])

services are the compute service, image service, object storage service, dashboard service, and identity service.

Each service is connected to others through a communication network or a solid line in Figure 2. The details of each service are the following and mainly from the OpenStack official documentation website [17]. The first is the compute service that is the box on the leftmost of Figure 2. It can enable a provider to control an IaaS cloud. This service can also allow the provider in controlling over instances or VMs and networks and in handling access to the cloud via projects and users. It also defines drivers that interact with underlying virtualization mechanisms or a hypervisor that is run on the provider's host OS or dom0. This service also publishes its functionality for other services via a web-based Application Programming Interfaces or APIs. The compute service is usually designed to use the image service. The second is the image service that can store VM images and manage a list of accessible images. A VM image is a file that can be instanced then became a VM or domU.

The third is the object storage service that is used for scalable, redundant data storage using clusters of standardized servers to store petabytes or PB of accessible data. This service is a long-term storage system for large amounts of static data which can be updated and retrieved. The fourth is at the box on the top of Figure 2 or the dashboard service. It is a web-based interface that allows a provider to manage OpenStack resources and services, and to interact with the compute service cloud controller using the OpenStack APIs. The last service is the identity service that is the default identity management system for OpenStack. After the IaaS provider installs this service, the provider can configure the service and can also initialize data into this service.

2.3. **The logging system in OpenStack.** Our previous work [8] illustrated the IaaS architecture and logging system architectures. Section 2.2 also illustrated the OpenStack architecture. However, no research illustrates how IaaS and logging system architectures can be applied to or combined with the OpenStack architecture. This section briefly introduces this combination, which is the innovative part of this work. The examples of the usefulness of this combination are: i) to be used as a fundamental to construct a real-world IaaS OpenStack cloud with the logging systems and with more than one compute node as will be discussed in Section 3.2.3, and ii) to be alternative solutions apart from OpenStack security features to mitigate risks associated with the CSA top severe threat as will be discussed in Section 4.2.4. The full details of the combination are in Section 3. [8, 12] discuss the security concerns in a public cloud on issues of confidentiality, integrity, and availability that relate to the security issues in the CSA report [13]. This report indicates an effect on the provider and customer on a public cloud. From the security concerns, [8] proposes a method to mitigate risks associated with CSA threats for an IaaS public cloud by using logging systems. [8] describes the logging system for IaaS as a system in the cloud infrastructure of the provider that collects and stores logging data of infrastructure's components such as VMs or domUs. A logging system consists of a logging process and log file. The logging process is responsible for recording data, whereas the log file used for storing data obtained from the logging process. In Figure 3, consider only the part of the compute node or a dotted square box on the right of this figure, the compute node is a computer machine with a hypervisor. A logging system is inside

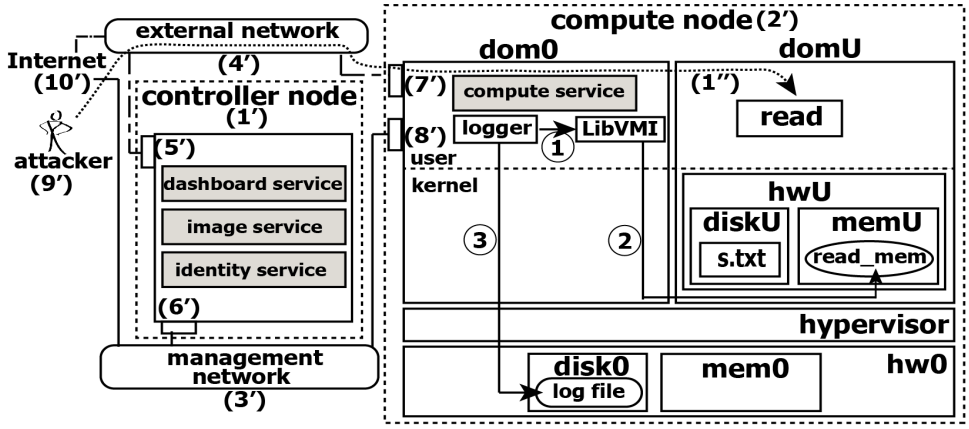

FIGURE 3. The architecture of the logging system in the OpenStack for IaaS

the compute node in the OpenStack environment. Section 2.1 described each component of IaaS architecture. This section will describe the additional components related to a logging system. In the compute node in Figure 3, any word ending with 0 indicates that this word or component is physically owned and managed by a provider, and ending with U indicates that the component is virtually owned and managed by a customer. Hw0 is a physical computer to be used as an IaaS VM host system, the same as hw discussed in Section 2.1. HwU is virtual hardware of a domU. Disk0 is the physical disk of dom0, and diskU is the virtual disk of domU. Mem0 is the main memory of hw0 or dom0 and menU is virtual main memory of hwU or domU. In Figure 3 see the logger box in the user level in dom0, it is a logger or logging process of a logging system that is run in dom0 to capture logging data. In Figure 3 see the box in the user level of domU, it is read or read process. In the experiment, we assume that this process can maliciously read IaaS customer's sensitive file or s.txt, see the box inside diskU. This file is normally stored in diskU that is owned by the customer or the owner of domU. This file can be an asset and valuable for a business organization of a customer as argued in [8]. Finally, the compute service is an application in the user level of dom0 for controlling the hypervisor, which is not related to the logging system.

3. The Experiment.

3.1. The aim of the proposed logging system in the OpenStack environment for IaaS.

[8] proposed logging system architecture to mitigate risks associated with CSA threats for IaaS. The main components of the architecture are the logging system or logger in Figure 3 and log file in the same figure. This paper defines the appropriate locations for placing the logging system components in the OpenStack environment to detect the target process or the read process on the IaaS customer's domU. We focus on the compute service, see the box on the leftmost of Figure 2. [18] states that the compute service is the most vulnerable part and the main component of the OpenStack. It is also a channel that leads to other critical IaaS resources such as VM or domU in Figure 3. For processes that are detected, it is a process which we suspect that the attacker will use it to access customer's sensitive files in diskU.

3.2. The system architecture of the proposed logging system in the OpenStack environment for IaaS in the experiment.

3.2.1. *The IaaS architecture on public cloud in the OpenStack environment.* In Figure 3, we illustrate the perspective of the IaaS architecture in the OpenStack environment, for short the IaaS OpenStack cloud. We run Ubuntu 16.04 LTS OS on a computer or hw0 as the compute node. Then we install OpenStack Ocata version in this OS. We map each

service of the OpenStack architecture in Figure 2 onto two physical computers. According to the function in the installation manual [19], one of these computers is the compute node, and the other one is the controller node. See the dotted box with number 1' on the left side of Figure 3; it is the controller node for execution of the three services of Figure 2, including identity service, image service, and dashboard service. These services were described in Section 2.2. See the dotted box with number 2' on the right side of Figure 3; it is the compute node execution of the compute service from Figure 2. This service is used to control hypervisor or Xen to manage dom0 and domUs. The object storage service in Figure 2 is not related to this experiment. There can be many of these nodes in the real-world environment. However, to simplify the experiment, from the architecture mentioned above or Figure 3, we use only one controller node, and one compute node. Both of the nodes use two network adapters (5' to 8') to connect the external network (4') and the management network (3'), respectively. See the dotted line with number 1", the external network is used for the customers to connect to domUs via the Internet or 10', in this IaaS OpenStack environment. In the experiment, this external network may also be somehow a channel that attacker or 9' can use this channel to connect to domUs with the credentials and passwords that the attacker may steal through phishing and fraud as argued in [8]. The management network is used for installing applications and for updating the patch of the security of the operating system of the controller node and the compute node. After building IaaS by using OpenStack or we called it the IaaS OpenStack cloud for the experiment, we got our IaaS OpenStack cloud. Then, we simulate incidents that were assumed to be taken place by the attacker. Thus, the attacker can connect to a domU in this built IaaS OpenStack cloud over the Internet using the credentials and password mentioned above. Then, the attacker may use read application or process (see the box in the user level of domU in Figure 3) to access to the sensitive file in diskU. We aim to locate the existing logging system or the logger (see the logger box in the user level of dom0) and the log files into this built IaaS OpenStack cloud. So, this logger has been designed to detect read application (read process). We consider the read process as a target process that the logger wants to capture the appearance of this process for the detection of such incidents.

3.2.2. *The architecture of the proposed logging system in the OpenStack environment.* Figure 3 is mapping our existing logging system architecture onto the built IaaS OpenStack cloud environment. Thus, the internal of the compute node in Figure 3 is the proposed logging system architecture and the main components of this logging system that were described in Section 2.3. In this research, we focus on the compute node because it is the most vulnerable part and the main components of the OpenStack (which has given the reason in Section 3.1). In this section, we illustrate the locations of the logger and log file of the proposed logging system in the built IaaS OpenStack cloud. There are three main components: the logger, LibVMI, and the log file. The LibVMI is a library to read data from memU in domU. The steps of the logger were discussed in our previous work [8] as the following. The first step or number 1 in Figure 3, see the logger box in the user level of dom0, the logger calls the LibVMI library to read data from read_men in memU. The second step, see number 2 in Figure 3, the library retrieves appropriate logging data from read_mem. The data is generated when the read process is executed. The examples of the data are the name of a process such as 'read' and Id of the process such as '1265'. See number 3 in Figure 3 the third step, the logger stores the captured data from step 2 into the log file in disk0.

3.2.3. *To construct a real-world IaaS OpenStack cloud with more than one compute node.* The proposed architecture in Figure 3 can be expanded to construct a real-world IaaS OpenStack cloud with more than one compute node. We also believe that our logging system (the logger, LibVMI, and log file in Figure 3) should also be ready to work in a

production environment (more than one compute node). We will discuss this in Section 4.2.2. [15] states that OpenStack can support a large number of compute nodes and storage units. Thus, in an OpenStack environment, a real-world IaaS OpenStack cloud is when this cloud is at least constructed with many compute nodes, not with only one compute node that will be discussed in Section 4.2.1. When constructing a real-world IaaS OpenStack cloud with our logging system, this can be done by step 1 or duplicating the whole compute node in Figure 3 with all its components including the logger, LibVMI, and log file. In step 2, we can connect the new duplicated compute node obtained from step 1 to the controller node or the dotted box at the left of Figure 3. Then, this cloud was extended from one compute node to two compute nodes and become a real-world IaaS OpenStack cloud. Both steps can be continuously repeated to construct a real-world IaaS OpenStack cloud with our logging system. Both steps are based on the builders or providers requirements on how many compute nodes in this IaaS cloud they need. This is our future work and out of the scope of this paper.

4. The Results and Discussions.

4.1. The results.

In Figure 4, the line with number 1 is when the logger wants to capture the name and the Id of the monitored process or read inside domU. Thus, the logger command or 'logger' in the box with 'a' is executed in dom0 by the provider. Also, there are two parameters of this command including the name of domU or 'instance-00000007' (see in the box with 'b') and the name of the monitored process or 'read' (see in the box labeled 'c') on domU. The logger command will check read_mem until it found the read process has appeared in read_mem. When the read process is executed in domU or see the box in line with number 1 in Figure 5, this is when the name and Id data of read process have appeared in read_mem. Then, the logger captures the Id of read as '1265' and the name of read as 'read', see the boxes with 'a' and 'b' in line 2 of Figure 4, respectively. Then, the logger command will be terminated.

FIGURE 4. The logger in dom0

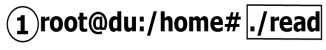

FIGURE 5. The read command in domU

4.2. Discussions.

4.2.1. *Our previous logging system working in an IaaS OpenStack cloud with one compute node.* Although the logging systems in our previous work [8] can work well in the laboratory environment or Figure 1, we never apply the systems to a real-world IaaS cloud like in an IaaS OpenStack cloud or Figure 3. After the experiment to apply our existing logging system to an IaaS OpenStack as illustrated by Figure 3, the results of the experiment in Figures 4 and 5 in Section 4.1 is the proof that our existing logging system from our previous work [8] can be applied to and work in the new built IaaS OpenStack cloud or Figure 3. Thus, this logging system in Figure 3 has the essential abilities to detect and record a monitored or malicious process caused by the CSA threat. Then it can store the captured data or logging data into the log file. The code of logger command of the logging system in Section 4.1 can also be modified to detect the new process and any file

that the new process maliciously reads or performs other operations with the file such as maliciously deleting the file as agreed in [8, 10]. This modification is out of the scope of this paper. However, we believe that our proposed logging system architecture or Figure 3 could be applied to and works in any IaaS cloud that is built by OpenStack. Then, this logging system can be one of the solutions to help in mitigating the risks associated with the CSA threats in a real-world IaaS OpenStack cloud production. Moreover, Section 3.2.3 already discussed to construct a real-world IaaS OpenStack cloud with more than one compute node.

4.2.2. *Why the existing logging system can still work in a real-world IaaS OpenStack cloud with more than one compute node.* There is only one compute node in the experiment or Figure 3. However, (when one desires to increase the number of compute nodes from one (in the experiment) to two or more to build a real-world IaaS OpenStack cloud as just discussed in Section 3.2.3 above, and from the experiment or Figure 3 and the results from Figures 4 and 5), we believe that our logging system can still work in this new built real-world IaaS OpenStack cloud with more than one compute node. The system can also still detect and record a monitored or malicious process, as discussed in Section 4.2.1. This is because of that when constructing a real-world IaaS OpenStack cloud with more than one compute node as discussed in Section 3.2.3, increasing the number of compute nodes by adding a new compute node into this newly built real-world IaaS cloud does not affect the operations of the existing compute node or the dotted box at the right of Figure 3. Also, it is because of that each compute node in this newly built IaaS environment is independent of one another. Moreover, a logging system in the new added or duplicated compute node will have the same functions as the logging system in the previous compute node in Figure 3. Thus, we believe that our logging system in Figure 3 which has the log file that is produced from the logger and LibVMI can still mitigate risks associated with the CSA threats in a real-world IaaS OpenStack cloud in a production environment, not only in the simulation or laboratory environment, as also argued in [8].

4.2.3. *Our logging system compatible with a variety of hypervisors.* Our logging system in Figure 3 can also be compatible with a variety of hypervisors. [15] states that OpenStack can support a variety of hypervisors such as VMware Elastic Sky X or ESX, Microsoft Hyper-V, Linux Containers or LXC, Kernel-based Virtual Machine or KVM, Quick Emulator or QEMU, Xen, and XenServer. Thus, this makes OpenStack to be used to build an IaaS cloud with a variety of hypervisors. In the experiment, the hypervisor of our compute node in Figure 3 is only Xen. However, [20] states that a hypervisor of a compute node can be changed from Xen to be KVM or QEMU, as also discussed above. Moreover, our logging system or the logger in Figure 3 works with LibVMI that is compatible with and can work with KVM or QEMU, not only with Xen, as agreed in [21]. Thus, our logger can still work to detect and record a monitored or malicious process (discussed in Section 4.2.1) in both new KVM or QEMU environment, not only in Xen. This can expand our logging system to work in at least two more hypervisors or KVM and QEMU of IaaS clouds which are built from open-source software such as OpenStack. This software is increasingly used in the IaaS cloud ecosystem, as agreed by Red Hat company in its report [14]. The company also offers its OpenStack version called 'Red Hat OpenStack Platform' to its clients such as Cathay Pacific. Thus, our logging system should also work with ESX, Hyper-V, LXC, and XenServer.

4.2.4. *The proposed architecture as alternative solutions apart from OpenStack security features to mitigate risks associated with the CSA top severe threat.* As discussed in Section 4.2.1, the code of our logger can be modified to detect when a malicious process reads a sensitive file that is belonged to a domU. The CSA [13] states that the first or most severe threat is data breaches. A data breach is when incidents have occurred with sensitive

data in an IaaS cloud and even the other types of the cloud such as Platform as a Service or PaaS [13]. This sensitive data in the IaaS cloud (such as health and financial information) should not be exposed, stolen, or used by an unauthorized user [13]. OpenStack is an open-source software to create an IaaS cloud. The data in this cloud may also be affected by this CSA thereat discussed above. When building an IaaS cloud by OpenStack, we will call this cloud as 'an IaaS OpenStack cloud' in this discussion. [22] agrees that an IaaS OpenStack cloud already had three features to maintain data security of the data of this cloud. However, these features may still yield disadvantages, as will be discussed one by one here. The first feature is the key management which helps to manage five mechanisms to maintain the confidentiality of the data in a newly built IaaS OpenStack cloud. This feature may decrease the performance of a key management server of this cloud. The reasons are the following. We will discuss only the first two mechanisms in this paper for the purposes of this discussion of decreasing the performance. The first mechanism is the key generation that can produce keys to be used to perform cryptographic of the data in an IaaS OpenStack cloud. The second mechanism is storing and retrieving the produced keys that are used to perform encryption and decryption, and certificate generation of this data in this cloud. [22] agrees that both mechanisms mentioned above face mainly two issues. The first issue is when too many simultaneous requests from computers of this cloud's customers via the Internet to a key management server. These requests also need many key generating and key retrieving mechanisms simultaneously. This may cause bottlenecks of this key management server. The second issue is that the key generating mechanism is an intensive computation process on the key management server.

Thus, the first feature to maintain data security of the data in an IaaS OpenStack cloud which relies on the key management may decrease the performance of the key management server of this cloud. The second feature is the block storage encryption, that is the encryption and decryption of a block storage unit. A block storage unit is a method of storing the data in a storage-area network (SAN) environment. An IaaS OpenStack cloud needs to deploy SAN, and the data of this cloud is stored in a volume or block. The block storage encryption operates in the compute node or the dotted box at the right side of Figure 3. Operating encryption on the compute node may yield disadvantages. This is because the block storage unit cannot apply the advantage of the compression function. It is because this function is one of the keys functions to reduce 90% of data in the provider storage [22]. Without this function, the provider may need to invest more budget for its storage units. The final feature is called the image integrity that can check the integrity of an image file of a VM or domU whether this file is modified by malicious code or compromised by an attacker or not. (A VM image is a file that can be instanced then became a VM or domU.) This file is in the controller node or the dotted box at the left of Figure 3. However, Benjamin et al. [22] measured overheads of the image integrity feature by the length of time required to launch a domU with and without signature verification. Then, they found the launch time of an image file with a signature verification higher than another launch time without the signature verification. It is because this higher launch time needs a massive computing process for the secure hash algorithm of the image file, as a part of the signature verification [22]. Thus, this feature may reduce the speed of overall operations in an IaaS OpenStack cloud. Then, this may also slow down the activities of domUs in this cloud infrastructure. An IaaS OpenStack cloud already had three features to maintain data security of the data of this cloud. However, these features may still yield disadvantages. Our logging system in an IaaS OpenStack cloud can be alternative solutions apart from OpenStack security features to mitigate risks associated with the CSA top severe threat.

4.2.5. *The proposed architecture without encryption and decryption mechanisms and with a lightweight computation.* Sections 3.2.3, 4.1, 4.2.1, 4.2.2, and 4.2.3 discussed how our

existing logging system from our previous work [8] can perform in a real-world IaaS OpenStack cloud, see in Figure 3. This system has abilities to detect and record a monitored process in or customer's critical files in a domU. Then, it can store the captured data or logging data into the log file. The contents of the log file can be analyzed or checked for examining any incident that may occur, as also done in [8, 10]. Thus, we believe this logging system can be one of the solutions to mitigate risks associated with the CSA threat one. This logging system in this paper can usually work without cryptographic techniques not with these techniques as done by three features to maintain data security of the data in the OpenStack environment as just discussed in Section 4.2.4 above. Then, this should allow our logging system to have a lightweight computation.

5. **Conclusions.** An Infrastructure as a Service cloud can offer rentable virtual machines or VMs to customers. However, CSA indicates that there are threats of an IaaS such as confidentiality, integrity, and availability of customers' files in the VMs. Thus, for the aim to mitigate the risks associated with the most severe CSA threat or threat one (the example of threat one is when an IaaS customer's file is unauthorizedly viewed), this paper applied our previous logging system into a new IaaS environment. A logging system can be one of the solutions to mitigate these risks. The system can capture activities of a malicious process that is reading a customer file in a VM. However, logging systems in our previous work, such as [8] cannot work in a real-world IaaS cloud that is built from widely used OpenStack software. We call this cloud an IaaS OpenStack cloud. Our logging systems can only work in a simulation or laboratory environment. Then, this paper successfully applied a previous logging system to an IaaS OpenStack cloud. We also found many useful aspects of the application of this logging system in this paper, such as the system can work with the other virtualization softwares such as Kernel-based Virtual Machine or KVM rather than only Xen. Thus, this paper can be an alternative solution apart from three OpenStack built-in features that are used to maintain data security of an IaaS cloud which is built from OpenStack. Then, this paper could truly enable our logging systems from our ten years of work [5, 6, 7, 8, 23] to be integrated and compatible with the widely used OpenStack software. This can help to mitigate risks associated with CSA threat one in real-world IaaS cloud production, and maybe in other types of the cloud such as Platform as a Service or PaaS. This paper also can be a guideline for mitigating other critical CSA threats such as the second most severe threat or 'Insufficient Identity, Credential and Access Management'. Our future work can be to measure the performance of the logging system of this paper.

REFERENCES

[1] T. Grance and P. Mell, *The NIST Definition of Cloud Computing*, National Institute of Standards & Technology, Gaithersburg, MD, United States, 2011.
[2] *Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017*, 2017.
[3] H. Albaroodi, S. Manickam and P. Singh, Critical review of OpenStack security: Issues and weaknesses, *Journal of Computer Science*, vol.10, p.23, 2013.
[4] W. Runathong, W. Wongthai and S. Panithansuwan, A system for classroom environment monitoring using the internet of things and cloud computing, *Lecture Notes in Electrical Engineering*, vol.424, pp.732-742, 2017.
[5] W. Wongthai and A. van Moorsel, Logging system architectures for infrastructure as a service cloud, *Journal of Telecommunication, Electronic and Computer Engineering*, vol.9, nos.2-4, pp.35-40, 2017.
[6] W. Wongthai and A. van Moorsel, Quality analysis of logging system components in the cloud, *Lecture Notes in Electrical Engineering*, vol.376, pp.651-662, 2016.
[7] W. Wongthai and A. van Moorsel, Performance measurement of logging systems in infrastructure as a service cloud, *ICIC Express Letters*, vol.10, no.2, pp.347-354, 2016.
[8] W. Wongthai, *Systematic Support for Accountability in the Cloud*, Ph.D. Thesis, Newcastle University, 2014.

[9] P. Chan-in and W. Wongthai, Performance improvement considerations of cloud logging systems, *ICIC Express Letters*, vol.11, no.1, pp.37-43, 2017.
[10] P. Chan-in and W. Wongthai, Logging solutions to mitigate risks associated with security issues in platform as a service cloud models, *Information (Japan)*, vol.19, no.10, pp.4883-4890, 2016.
[11] C. Coles, *Cloud Market in 2018 and Predictions for 2021*, 2018.
[12] S. Ristov, M. Gusev and A. Donevski, OpenStack cloud security vulnerabilities from inside and outside, *Cloud Computing*, pp.101-107, 2013.
[13] C. S. Alliance, *The Treacherous 12: Cloud Computing Top Threats in 2016*, 2016.
[14] redhat, *Digital Transformation – The Open Source Way*, 2017.
[15] O. Sefraoui, M. Aissaoui and M. Eleuldj, OpenStack: Toward an open-source solution for cloud computing, *International Journal of Computer Applications*, vol.55, no.3, pp.38-42, 2012.
[16] OpenStack, *What is OpenStack?*, 2019.
[17] OpenStack.org, *Welcome to OpenStack Documentation*, 2019.
[18] I. A. Elia, N. Antunes, N. Laranjeiro and M. Vieira, An analysis of OpenStack vulnerabilities, *European Dependable Computing Conference*, pp.129-134, 2017.
[19] OpenStack.org, *OpenStack Installation Tutoial for Ubuntu*, 2017.
[20] G. Wang, Z. J. Estrada, C. Pham, Z. Kalbarczyk and R. K. Iyer, *Hypervisor Introspection: A Technique for Evading Passive Virtual Machine Monitoring*, USENIX Workshop on Offensive Technologies, 2015.
[21] B. Payne, *About the VMI Tools Project*, 2013.
[22] B. Benjamin, J. Coffman, H. Esiely-Barrera, K. Farr, D. Fichter, D. Genin, L. Glendenning, P. Hamilton, S. Harshavardhana, R. Hom, B. Poulos and N. Reller, Data protection in OpenStack, *International Conference on Cloud Computing*, 2017.
[23] W. Wongthai and A. van Moorsel, An approach to defining and identifying logging system patterns for infrastructure as a service cloud, *ICIC Express Letters*, vol.12, no.10, pp.1009-1016, 2018.