Towards Cloud Security Monitoring: A Case Study

Umar Mukhtar Ismail, Syed Islam, Shareeful Islam
School of Architecture, Computing & Engineering
University of East London
London, UK
u.ismail@uel.ac.uk, syed.islam@uel.ac.uk, shareeful@uel.ac.uk

***Abstract*— Cloud computing has become the norm in the provisioning of computing resources due to its flexible and proven reliability. Businesses perceive cloud services as a trend that presents enormous possibilities both in economic and technical terms. The growth in cloud services have also increased bottlenecks and security risks to business assets. Cloud security monitoring has remained relatively unexplored in security terms, a factor that has led businesses to be oblivious on the metrics to capture and the appropriate techniques to use.**

In this paper, we explore security monitoring in terms of tracking specific user requirements based on a case study. We identify various security tools that are practically relevant for addressing the requirements, and devise selection criteria for choosing the best tools. We present an evaluation of the tools and present a ranking for the tools that meet the particular requirements of the case study. The effort in this paper broadens the notion of cloud security monitoring and provides a methodical practical approach to solving a security related issue.

Keywords — cloud computing; monitoring; security monitoring.

I. INTRODUCTION

Cloud computing services have over the years become the game changer that provide businesses with enormous opportunities that were simply unavailable before. Large and midsize businesses are increasingly leveraging cloud services to improve the quality of services delivered to end-users and reduce cost because cloud environments are designed with scalable components that ensure continuous service delivery. The aspect of scalability is largely associated with lower cost of ownership and opens up diverse possibilities for users to improve productivity, expand earnings, and reach out target markets [1].

These same range of possibilities imply surrender of control over physical security, sharing resources with cotenants and lacking the knowledge of where cloud resources are hosted [2]. In a nutshell, user assets become vulnerable to various security issues and challenges that are exacerbated by the lack of monitoring capabilities and decreased visibility into the security status of assets being hosted by the Cloud Service Providers (CSPs) [3, 4]. Nevertheless, users have over the years gained appreciable understanding of the need to monitor assets and the threats associated with the benefits of cloud computing, and to some extents, methods have been devised by CSPs to effectively provide users with monitoring capabilities so as to increase the adoption of cloud services. For instance, CSPs offer dashboards for tracking service availability, timely discovery of service outages and performance metrics [5]. These attempts by CSPs although sufficient for performance and functional requirements, cannot be said to be sufficiently convincing for end-users to rely on, particularly whose proclivity is the monitoring of security related metrics.

The work in this paper seeks to unravel the issue of tool-based (practical) cloud security monitoring by discerning the approaches and tools for enabling users attain security visibility in the cloud. It augments existing literatures in the area of cloud security by using a systematic approach that reflects on real-life requirements to help cloud users address one of the most pressing concerns associated with gaining visibility. We present a real-world case study to evaluate the ability of the tools and approaches to meet the security requirements based on published properties and functionality. We believe that this paper contributes to existing work by forming the first step to creating a framework for identifying cloud security monitoring requirements and implementing solutions to meet such requirements.

The paper is structured as follows: section two discusses related works and section three covers security requirements. Section four presents monitoring in general sense while identifying its implications for cloud. Section five presents existing approaches and tools that can be employed for cloud monitoring and section six presents the case study. Section seven covers the evaluation, while section eight draws conclusions and outlines future work.

II. RELATED WORK

There are several existing works in the area of cloud monitoring. Alhamazani et al [6] reviewed commercial cloud monitoring tools by considering applications within the various cloud layers. Aceto et al [7] analysed the properties that are key to monitoring cloud systems and adopted a methodology that analyses state of the art cloud monitoring techniques. Similarly, Fatema et al [8] identified essential features that are desired for operational monitoring in the cloud, reviewed and analysed a broad range of monitoring tools that are being used to observe cloud functional resources. Krizanic et al [33] performed a review and categorisation of monitoring tools according to Operating Systems (OS), notification and other services being supported by the cloud, while Rimal et al [34] presented a taxonomy of cloud services based on comparative study of different CSPs and their systems.

Although, our work does not include a systematic literature review of the tools, it is a significant contribution and differs from other literatures by using a case study to reflect on how the tools can fulfil real-life monitoring requirements. It also

focuses on cloud security monitoring in addition to a selection criteria of suitable tools.

III. Security Requirements

Security is an essential property of every information technology environment including cloud systems, which plays an important role of ensuring the protection of assets from a wide range of threats [2, 3, 9]. Within the context of information systems, security requirements could be described as a subset of nonfunctional requirements that relates to ensuring the *confidentiality*, *integrity* and *availability* of assets [10]. Confidentiality deals with providing assurance that an asset is not accessed or disclosed to unauthorized processes, systems or persons. Integrity involves the assurances of preventing unauthorized systems, processes or persons to modify, create or delete an asset. Availability on the other hand is concerned with ensuring the ability of users and applications to access an asset at any time and in the expected format. These security requirements constitute the inherent expectations that must be met by any deployable control to avert risks and they can be referred to as a foundation or a baseline when evaluating the efficiency of monitoring systems. In the following section, we intend to use the concepts of confidentiality, integrity and availability in classifying user security requirements and the ways in which monitoring systems fulfill them.

IV. Monitoring in Cloud

Monitoring is an important process that systematically collects information regarding current state or extracts information from a log of past events that occurred in any component of a cloud service [11]. The role of every monitoring process in cloud infrastructure serves to collect, process and report information on metrics such as performance or other custom metrics against well-defined behaviour rules or policies [6, 7]. An important attribute of monitoring system is to constantly run on any underlying infrastructure in a transparent manner for the detection of faults, bottlenecks and other phenomenon and perform automatic actions or generate reports for resolutions. A strategic cloud monitoring process usually starts by identifying all the components or actors authorised to access a resource, the categories of resources to be monitored and a logging solution that records every event. Identifying the actors could be an easy task, perhaps what could be more intricate is to identify where to position monitoring agents with the task of gathering primitive data related to predefined metrics within different cloud layers [12]. Additionally, there are various areas in the cloud that attract monitoring needs such as data centre operations, performance, billing, and service availability. Nonetheless, security monitoring is considered as more crucial because of security being a special consideration to cloud adoption [2]. In the following section, discussions are narrowed and emphasis given to cloud security monitoring.

A. Security Monitoring In Cloud

Security monitoring in cloud is a dynamic continuous activity that involves effective and proactive management of cloud components in order to identify and respond to threats and vulnerabilities within a cloud service [12]. Security monitoring incorporate various systems that are designed to accept event logs from multiple systems and provide relevant data output which can be correlated and analysed. A robust architecture exclusively utilizes the dynamicity of cloud properties and collects data in accordance with pre-established metrics, rules and policies through a broad range of security systems [13]. These security systems, including technologies such as intrusion detection/prevention systems, firewalls, and other solutions ensure the logging of information required to successfully strengthen transparency, visibility and eliminate uncertainty surrounding cloud services.

Other key aspects of security monitoring involve processes for collecting and analysing data for events of interest to automatically respond to security-related incidents, as well as providing a powerful structure to predict future security related issues. A well-structured cloud security monitoring supports timely dissemination of security occurrences to interested actors for decision on the appropriate course of action that ought to be taken. From the side of cloud customers, security monitoring plays an important role of enhancing accountability and mutual trust by means of making CSPs accountable for security violations that emerge as a result of deficient controls in their services, help customers detect a breach of Service Level Agreement (SLA) contracts, and help in the gathering of evidences to validate the security claims of a CSP. On the part of a CSP, it empowers the capturing of current security state of cloud systems, deviations from expectations and the monitoring of clients' activities to ensure malicious use of resources are prevented.

B. Challenges to Security Monitoring in Cloud

The task of security monitoring in the cloud is more complex owing the fact that information of different granularity is aggregated from heterogeneous components dispersed across multiple levels at different time intervals. In our view, challenges to cloud security monitoring include:

- *Heterogeneous nature of the cloud* - The diverse nature of cloud computing creates an environment that is complex and difficult to monitor, where traditional and cloud specific monitoring tools might not render the scalability required to monitor the state of hosted resources.
- *Systems and devices to be monitored* - Security monitoring might not be designed to cover every cloud systems or devices. A lax in this aspect may omit certain systems from the monitoring architecture, which means that such systems may fail to log events and generate alerts on any attacks against them.
- *Difficulty in defining security metrics* - Monitoring rules and policies may not be comprehensively defined to support the detection of violations and prompt dissemination of occurrences to interested parties.
- *Log retention, access and storage* - Some monitoring functions such as anomaly detection adopt real-time event detection models, perhaps others involve log analysis. The challenge arises where logs are not retained in accordance

to appropriate retention policies or regulatory requirements that could support subsequent analysis of data. Another challenge deals with the storage and access to log data, especially in a third party environment like the cloud where deliberate or accidental interference might occur.

- *Integrated monitoring* - Users of cloud services usually opt to move some applications to the cloud while other applications are hosted in-house. The challenge in this respect is the ability to integrate monitoring and correlate data from different environments.
- *Constrained SLAs* - CSPs may not be willing to deliver acceptable technology tools that enable sufficient monitoring of security metrics defined in SLA or even support independent monitoring tools that allow customers to monitor actual security conditions of both their assets and the cloud environment as a whole.
- *Positioning of probing agents* - This involves identifying the appropriate location within the cloud layers where probing agents could interpret and execute an underlying policy to detect events of interest.

Thus, in order to address some of the challenges and achieve monitoring objectives, it is imperative to consider the different layers of cloud infrastructure so that the target where monitoring data is captured can be identified.

C. Layers of Security Monitoring in Cloud

Security monitoring specific to cloud must be able to deploy on a specific or multiple cloud components, collect and disseminate monitoring metrics to subscribed actors [14, 15]. The components are deployed within multiple layers such as network layer, OS layer and virtualization layer. It is imperative to consider these layers from the perspective of a cloud customer for a better coherence of the relevant layers where security metrics can be collected, processed and reported within a cloud using different tools.

- *Hardware Layer* - Hardware layer in cloud stack integrates various physical resources that underpin cloud services. Resources such as servers and networks are deployed and maintained by the CSP at this layer and monitoring aims to collect important data about the status of physical infrastructure. Monitoring tools and services are deployed by the CSP for identifying issues and events that may affect custom metrics. However, monitoring data are usually not shared with customers, yet a CSP is expected to provide security guarantees and compliance with relevant regulatory frameworks.
- *Operating System (OS) Layer* - This layer deals with the basic software component that manages operations, executions, processes of hardware and software resources both on the host and guest machines. It supports the execution of functionalities of other layers, including components that a cloud service relies on to provide other capabilities. Monitoring at this layer essentially tends to focus on metrics relating to system files, however, the choice of metrics may vary according to other monitoring goals. The monitoring models are normally set by the CSP, especially in Infrastructure-as-a-Service (IaaS) model where either the CSP provides the main host OS and customers control the guest OS. This trend makes monitoring more flexible and responsibility can be shared between the CSP and a customer, depending on the SLA agreement.
- *Network Layer* - The delivery of cloud services is performed over distributed networks at different levels and protocols. This layer plays a fundamental role of supporting the connection links between cloud components and between the cloud service and end users. The monitoring of different network solutions is performed using various tools located at different network segments. The deployment of monitoring tools could fall under the responsibility of the CSP or the customer. For instance, in IaaS model, the CSP is responsible for managing and monitoring the physical and data link levels, while the customer is responsible for network levels.
- *Virtualisation Layer.* Virtualisation sits between the hardware and the OS. It enables the partitioning of resources of a physical system into multiple virtual resources. It is the main enabler of cloud computing that allows pooling of resources and serve multiple users using multi-tenancy. Multi-tenancy allows users to share the same physical infrastructure by assigning users virtual resources that run on top of the physical infrastructure. The cloud users are usually responsible for enforcing monitoring tools at this layer, to monitor and ensure secure communication between various system components by ensuring that all virtualised resources transmit and receive data in encrypted formats and protected against malicious code manipulations.

V. Approaches to Cloud Security Monitoring

It is eminently difficult to develop a security monitoring tool that can effectively support evidence collection, analysis and generate output from all cloud systems, especially if done in real time. One reason for this could be the complex interlaced structure of various components within different layers of the cloud stack and the evidences that ought to be collected. To reify an effective approach, it is essential to focus on a particular component and then define the variables, logs and metrics that need to be collected, the tools that can be used for the collection, how alerts/alarms are generated, and what action is executed after an alarm. This can be achieved by understanding the policy and operational structures of the business and asset requirements. Only when these factors are considered would a monitoring technique be effective in addressing security related metrics. In general, there are several ways in which security monitoring can be performed as shown in Table I.

TABLE I. Security Monitoring Approaches

APIs	Cloud Monitoring Systems	OS Based	Commercial Monitoring as a Service
Custom Built API	*Open Source*	*Command Line*	*CSP Based*

Prebuilt & Modified API	*Commercial*	*File System*	*Thirdparty*

We contend that there are essentially four types of approaches that can be adopted for security monitoring as shown in Table I: API, cloud monitoring systems, OS, and specific monitoring as a service. Each of the approaches entails a number of supported tools that can essentially effectuate security monitoring objectives. In this respect, we perform a brief review of a few selected tools under cloud monitoring systems and OS based tools. The rationale behind doing this is to give a flavour on the variety of existing toolkits that can be used for security cloud monitoring. Cloud as a service are not covered in this work by reason of licensing considerations.

A. APIs

Using this approach, there are three fundamental choices that can be adopted. The first involves the design of a custom software that adapts to specific requirements and security monitoring of any kind. In developing a custom monitoring software, a middleware API, for instance, is created and every application, program or system request is logged while passing through the API. It can then be utilized to collect desired metrics for analysis. The second is to use a prebuilt monitoring tool that collects data on metrics relevant for user requirements. The third is to modify an existing tool by incorporating additional plugins to become suitable for user needs. This is more likely to be relevant in situations where a prebuilt tool lacks basic security functionalities to satisfy specific user requirements. A well-known API framework that can be used for such request logging and monitoring is JADE[16]

B. Cloud Monitoring Systems

This approach entails systems and tools that are specifically designed for tracking and managing the operational workflow of system operations, applications, networks, and in some cases, user activities in a cloud service. Tools in this category are generally implemented through automated monitoring process that provide centralised view over security and management. It is important to note that such systems may not be specifically tailored to security metrics, however some of them contain modules that could be extended or defined differently to capture security metrics that are useful.

- *Private Cloud MONitoring Systems (PCMONS).* Focuses on VMs by retrieving, gathering and preparing relevant information for monitoring data visualisation based on a three layered monitoring architecture. It integrates with other monitoring solutions such as Eucalyptus for IaaS and Nagios for enabling visualisation of monitoring data. For integration, it comprises modules such as node information, cluster data integrator, monitoring data integrator, VM monitor, configuration generator, monitoring tool server, user interface and database [17].

- *OpenNebula.* A management toolkit for virtualised datacentres that provides different features at two layers in data centre virtualisation management and cloud management. The cloud management supports a provisioning layer for infrastructure management solution and consists of modules to monitor physical infrastructures. Monitoring data are collected through different category of probes that are installed on the nodes and are queried through SSH connections [18].

- *Nagios.* Nagios is used for system and network monitoring but can be extended to support monitoring capabilities suitable for variety of servers and OS in cloud computing infrastructures. It comes in three versions – Nagios XI, Log Server and Network Analyser. Nagios XI monitors cloud infrastructure components such as applications, and network infrastructure. The Log Server gives the ability to set up alerts to notify users of potential threats as well as system audit tasks. The Network Analyser inspects network traffic for potential security threats to provide users a highly granular data for network analysis. All the versions are adoptable for monitoring virtual machine (VM) instances on OpenStack and Amazon services [19].

- *Hyperic HQ.* Helps in the improvement of infrastructure availability and consists of streamlined management modules for software and network resources monitoring. The first module is a discovery module that deploys agents on a machine to automatically discover all software resources running on a machine and collect key facts about. The other component, monitor, gives options to select the metrics for which data are captured and reflect in the dashboard. Alerts are set on the selected metrics and it can respond in a variety of ways by sending email notifications or initiate a communication to another management system [20, 21].

- *SQRT-C.* This is a middleware scheme that is used for real-time resource monitoring of cloud deployments and is built based on OMG Data Distribution Service pubsub middleware. The design principles are based around three components: publishers, subscribers and managers. Publishers catalogue the nodes that are being monitored to operate a publisher which publishes state changes via a DDS protocol. Within the subscriber, a set of subscribers run closely to the publishers and monitoring state and enact decisions in real time, while the manager arranges connections between publisher and the subscriber [22].

C. Operating System Monitoring

This approach centres on monitoring specific security and other metrics in a cloud infrastructure OS, particularly for private cloud deployment (for Linux and UNIX). This approach also comprises two approaches i.e. the ability to monitor the host or guest OS. In both directions, monitoring detects predefined events associated with custom security metrics, file access, etc. There are numerous command line process and file monitoring tools that can be used for this purpose. A review is provided comprising a range of most commonly used command line resource and file monitoring tools that could be tailored for specific user requirements.

- *Top – Linux Process Monitoring.* Provides a dynamic view of actual process activity or running system in a real time. It is used to check CPU and memory utilisation process. Top also provides an interactive interface that can be used to manipulate processes and provides information regarding server availability including

uptime, memory usage, load average, and tasks status [23].

- *Lsof –Open Files.* List Open Files (Lsof) provides information about opened files and files opened by various processes running on a UNIX system. An open file may include a directory, character special file, network file, a regular file, a block special file, shared library, socket stream, etc. [24, 25] .
- *Psaact – Monitor User Activity.* Psaact is used for monitoring user activities on a system. It runs in the background and keeps track of user activities on a system as well as resources being consumed by users. It is useful for monitoring users of an application or data to observe what functions are being executed by users. Psaact not only informs the duration of time a user has worked on a file, but also the commands that were executed [25].
- *Nagios – Network/Server Monitoring.* This version differs from the cloud-based discussed above. It has specific features for in-house data centre and comes in three different versions: infrastructure monitoring, log server, and network analyser. Nagios XI for infrastructure monitoring provides monitoring capabilities for operating systems, applications, services, network infrastructure and system metrics in a centralised view. The log server provides the ability to create alerts from a web-interface based on queries and thresholds that are most important to users. Nagios network analyser performs an in-depth network traffic analyses to identify potential security threats [26].
- *M/Monit – Linux Process and Services Monitoring.* Monit is a monitoring and supervision tool which can perform various event-based actions to monitor programs, processes, directories, files and filesystems. It also provides the abilities to monitor changes to files and directories such as timestamps changes, checksum and size changes and send email notifications or take other responsive actions. It can be used to monitor daemon processes and similar programs that run on a localhost. It can monitor network connections to various servers in addition to performing network tests on a protocol level [27].
- *Security Onion – Security Monitoring.* Security Onion is more of a network security program for intrusion detection, network security monitoring and log management. It provides functions such as network and host based intrusion detection, packet capture and an analysis tool for managing and evaluating logs. Intrusion detection capabilities analyse both network traffic and host systems, generate log and alert data for detected events or activity. [28].
- *iWatch (Real time Filesystem Monitor)* - A file change notification system that provides an efficient way to trace actions in the filesystem in real-time. It monitors changes to a specific file or directory and send an immediate email notification once a change has occurred. It is very useful in keeping an eye on critical files or directory against unauthorised changes. iWatch comes in two different modes – daemon mode and command line mode [29].
- *Inotify (File System Activity Monitor)* - Used for monitoring individual filesystem operations such as read and write, and can also be used to monitor directories against a list of events. It sends an alert when an event occurs and enables tracking the origin and destination of an event. It has the capabilities to monitor any filesystem object, and when configured to monitor directories, it digs deep to reveal the name of the file inside the directory that has been changed [30].

D. Cloud Monitoring as a Service

Tools in this category are similar to the APIs discussed above, with the distinction that the backend operational processes, source code, and implementations are not made public. In other words, they are monitoring services that are designed by CSPs and accessed through APIs by users to track instances, monitor log files and set alarms on events of interest. Another class of monitoring as a service involves third party services that are external to the CSP. They tend to expand capabilities and to counterpoise what CSP tools cannot monitor. Both types of services provide basic functionalities that may come for free to cloud users, while custom functionalities usually attract substantial fee depending on the metrics desired. The proprietary nature of these services indisposes the consideration to determine their suitability for the requirements provided in the case study. Readers interested in this category may consider Amazon's CloudWatch [31], AzureWatch [32], etc.

VI. CASE STUDY

A. Description

The case study reflects an anonymized privately owned investment company. The company provides investors with active financial and investment management services. It connects with multiple stock markets that allow the tracking of market indexes in respect of sundry commodities and securities around the globe. In pursuit of structure and efficiency optimisation, and the customer extension and retention, and the need to more easily meet up with streamed activities

The current IT infrastructure used by the company comprises of three virtual servers. The first is the core server where trade transactions are processed and stored including company/clients' accounts, financial information, and used by employees to share information with stakeholders. The second is an administration server that enables basic administration functions such as setting up, editing and managing customer accounts. The third plays an important role of a proxy server that is configured to provide security for ensuring authorised access and legitimate use of resources.

B. Requirements

Senior-level stakeholders and primary decision makers in the company were engaged in discussions to highlight the key requirements associated with the assets. After numerous meetings, the security requirements and regulatory frameworks

applicable to the company's operations were analysed to determine the requirements that are key to its operations such as reliability, confidentiality, integrity, service continuity and compliance. In particular, specific essential requirements were identified using the framework in [33], which has also been validated by the stakeholders. They are categorised according to security requirements and presented in Table II.

TABLE II. REQUIREMENTS

Requirements	Details
Confidentiality	• Access to VM and user accounts. • All failed and successful access attempts to files and folders by company stakeholders, an attacker or CSP staff. • Folders and files being accessed by running application • Unauthorised copying and transmission of files and folders.
Integrity	• Unauthorised modification to VM and OS components • Unauthorised creation of VM and user accounts • Unauthorised changes to files and folders by legitimate and illegitimate users, including CSP staff and attackers. • Unauthorised modifications to folders and files by running applications.
Availability	• Unavailability of underlying OS and VM images • Disruption of cloud services • Unavailability of files, folders, applications and user accounts. • Continuous backup of critical data

C. The Security Monitoring Tools likely to sove the Case Study

In this section, we categorise and identify tools from Section IV that have the essential functionalities for monitoring all or some of the properties specified in Table II. It should be noted that the classification was incisively done in a general sense i.e. according to the overall consideration of the properties specified in the case study, rather than exploring individual properties and how they are supported by the tools.

Tables III and IV provide a taxonomy of the cloud monitoring systems and OS based monitoring tools respectively. The classification is done according to the tool's ability to meet the confidentiality, integrity and availability needs as specified in Table II. It is evident from these classifications that some tools support the essential features to satisfy all the monitoring needs, while others support less features. This trend is attributed to the fact that specific tools incorporate attributes suitable for cloud security monitoring, whereas other tools are neither specifically designed for security monitoring nor cloud computing hence they may be adapted for that purpose. In addition, it could be challenging to adopt a single monitoring tool or strategy that collects all of the properties, especially in real-time. The taxonomy evidenced this assertion by showing that many of the tools fail to provide complete solutions to the needs of the case study.

TABLE III. CLASSIFICATION OF CLOUD MONITORING TOOLS ACCORDING TO THE REQUIREMENTS

	Cloud Monitoring Systems				
Requirements	PCMONS	OpenNebula	Nagios	Hyperic HQ	SQRT-C
Confidentiality		×	×	×	
Integrity	×	×	×		
Availability	×	×	×	×	×

TABLE IV. CLASSIFICATION OF OS MONITORING ACCORDING TO THE REQUIREMENTS

	OS Based							
Requirements	TOP	Lsof	Psaact	Nagios	M/Monit	Sec Onion	iWatch	inotify
Confidentiality		×	×	×	×	×	×	×
Integrity			×	×	×	×	×	×
Availability	×			×	×			

D. Selection of Tools Suitable for the Case Study

The tools which are likely to fulfil the case study requirements are selected in this section. The selection criteria follow a rating scale using a global rating of 1 to 4, based on which tools with the most rating (between 1 and 2) are selected. The ratings are established as:

- *Rating 1:* All the requirements are fulfilled
- *Rating 2*: Most of the requirements are fulfilled
- *Rating 3:* Part of the requirements are fulfilled
- *Rating 4.* None of the requirements are fulfilled.

The traits considered in the scaling include the promised ability to fulfil the requirements. In Table V, a grouping of the tools with the most rating is provided to highlight those tools that achieved the highest rating.

TABLE V. RATING OF TOOLS BASED ON FULFILLING REQUIREMENTS

Rating	Tools
1	*OpenNebula, Nagios*
1	*Nagios, M/Monit*
2	*PCMONS, Hyperic HQ*
2	*Psaact, Security Onion, iWatch, iNotify*
3	*SQRT-C*
3	*TOP, , Lsof*

VII. EVALUATIONS

The activities performed in the previous sections classified the monitoring according to their potentiality of fulfilling the requirements, as well as a rating scale for selecting the suitable tools. Based on such assessment, tools with a rating between 1 and 2 were selected. Thus;

- *OpenNebula – L*ikely to support all the security requirements by providing powerful security management and high availability features. It supports modular and extensible architecture that is compatible with multiple platforms. It is configurable to deploy on public, private

or hybrid clouds and supports broad range of hypervisors such as VMware, Xen, KVM, etc.

- *Nagios* - Supports features to fulfil all the security requirements. It provides integrated capabilities, extensible architecture and various APIs that enable its features to be extended through the addition of plugins. In addition, capabilities support monitoring of applications, services OS, network components and other layers of the cloud.
- *M/Monit* - All the requirements can be fulfilled by M/Monit and it provides additional features of performing actions to fix a problem. It is configurable for UNIX systems, networks and cloud services, as well as having extensible attributes to monitor events from all monitored systems, making it an ideal choice for security monitoring in a private cloud.
- *iWatch, iNotify, Security Onion and Psaact* - Apart from availability requirements, all are utilities that provide the functions to fulfil confidentiality and integrity requirements as specified in the case study.

VIII. Conclusion And Future Works

The growth and the uptake of cloud computing calls for effective tools and strategies for security monitoring that enables the tracking of occurrences and events on user-specific needs. A considerable variety of tools offering different supports that are appropriate for security monitoring have long been proposed. An effective monitoring approach ought to provide fine-grained attributes for aggregating variables affecting security condition.

In this paper, we contribute to the area of security monitoring in general and cloud security monitoring in particular by providing a taxonomy that groups capabilities of different tools according to their potentiality to aid monitoring in cloud environment. In particular, the paper identified the approaches that can be adopted for security monitoring by providing a logical process to discover and select the most suitable tools that can help users attain the overarching tracking of crucial requirements. An important aspect of the contributions is the use of a case study to address security monitoring, which not only improves knowledge but also helps users facing the dilemma of determining security monitoring strategies.

However, a practical demonstration of the evaluated tools is not provided in this paper. A deliberation in this regard is intended for our future work, where practical implementation of the tools is performed in relation to the case study presented.

References

1. Schduten, E. *Five Cloud Benefits*. 2012 [cited 2016 03/06/2016].
2. Hashizume, K., Rosado, D., Medina F. E., Fernandez, E., *An Analysis of Security Issues for Cloud Computing.* Journal of Internet Services and Applications, 2013. **4**(5): p. 1-13.
3. Takabi, H., J.B. Joshi, and G.-J. Ahn, *Security and privacy challenges in cloud computing environments.* IEEE Security & Privacy, 2010(6): p. 24-31.
4. Wei, Y. and M.B. Blake, *Service-oriented computing and cloud computing: challenges and opportunities.* IEEE Internet Computing, 2010. **14**(6): p. 72.
5. Garg, S.K., S. Versteeg, and R. Buyya. *SMICloud: a framework for comparing and ranking cloud services*. in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. 2011. IEEE.
6. Alhamazani, K., Ranjan, R., Mitr, K., Rabhi, F., Guabtni, A., Bhatnaar, V., *An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art.* Computing, 2015. **97**(4): p. 357-377.
7. Aceto, G., Bhotta, A., Donato, W., Pescape, A., *Cloud monitoring: A survey*. Computer Networks, 2013. **57**(9): p. 2093-2115.
8. Modi, C., Patel, D., Borisaniya, B., Patel H., Patel, A., Rajarajan, M., *A survey of intrusion detection techniques in cloud.* Journal of Network and Computer Applications, 2013. **36**(1): p. 42-57.
9. Zissis, D. and D. Lekkas, *Addressing cloud computing security issues.* Future Generation computer systems, 2012. **28**(3): p. 583-592.
10. Mellado, D., E. Fernández-Medina, and M. Piattini, *A common criteria based security requirements engineering process for the development of secure information systems.* Computer standards & interfaces, 2007. **29**(2): p. 244-253.
11. Truong, H.-L. and S. Dustdar, *Composable cost estimation and monitoring for computational applications in cloud computing environments.* Procedia Computer Science, 2010. **1**(1): p. 2175-2184.
12. Ouedraogo, M., Dubois E., Khadraoui, D., Chenal B., *Adopting an agent and event driven approach for enabling mutual auditability and security transparency in cloud based services*. in *Proceedings of the International Conference on Cloud Computing and Services Science, Lisbon, Portugal*. 2015.
13. Krutz, R.L. and R.D. Vines, *Cloud security: A comprehensive guide to secure cloud computing*. 2010: Wiley Publishing.
14. Spring, J., *Monitoring cloud computing by layer, part 1.* Security & Privacy, IEEE, 2011. **9**(2): p. 66-68.
15. Ibrahim, A.S., Harris J., Grundy J., Almorsy M., *CloudSec: a security monitoring appliance for Virtual Machines in the IaaS cloud model*. in *Network and System Security (NSS), 2011 5th International Conference on*. 2011. IEEE.
16. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G., *JADE: A software framework for developing multi-agent applications. Lessons learned.* Information and Software Technology, 2008. **50**(1): p. 10-21.
17. Chaves, D., Uriarte, R., Westphall, C.,*Toward an architecture for monitoring private clouds.* Communications Magazine, IEEE, 2011. **49**(12): p. 130-137.
18. Sempolinski, P. and D. Thain. *A comparison and critique of eucalyptus, opennebula and nimbus*. in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. 2010. Ieee.
19. Nagios, *Cloud Computing and Cloud Computing Monitoring with Nagios*. 2016.
20. Hyperic, H., *System Monitoring Software*.
21. vmware, *VRealize Hyperic* 2016.
22. An, K., Pradhan, S., Caglar, F., Gokhale, A., *A publish/subscribe middleware for dependable and real-time resource monitoring in the cloud*. in *Proceedings of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management*. 2012. ACM.
23. Moon, S., *15 simple TOP command examples on Linux to monitor processes.* 2016.
24. Wallberg, S., *Finding Open files with Isof.* IBM, 2006.
25. Saive, R., *How to Monitor User Activity with psacct or acct Tools.* Tecmint, 2013.
26. Josephsen, D., *Building a monitoring infrastructure with Nagios*. 2007: Prentice Hall PTR.
27. Monit, 2016.
28. Burks, D., *Security Onion.* nd.[Online]. Available: http://blog. securityonion. net/p/securityonion. html.[Accessed 11 May 2016], 2012.
29. Wirawan, C., *iWatch: Realtime Filsystem Monitor.* 2011.
30. Streicher, M., *Monitor File System Activity with iNotify.* 2008.
31. Services, A.W., *Amazon CloudWatch.* 2016.
32. Window, A., *Azure Window.* Game of Thrones (TV series)-Unabridged Guide, 2012: p. 110.
33. Ismail, U.M., Islam, S., Ouedrago, M. *A Framework for Security Transparency in Cloud Computing.* Future Internet, 2016. **8**(1): p. 5.