2012 IEEE Eighth World Congress on Services

Validating Cloud Infrastructure Changes By Cloud Audits

Frank Doelitzscher*, Christian Fischer*, Denis Moskal*, Christoph Reich*, Martin Knahl* and Nathan Clarke‡
*Furtwangen University - Cloud Research Lab
Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany
Email: {Frank.Doelitzscher, Denis.Moskal, Christian.Fischer, Christoph.Reich, Martin.Knahl}@hs-furtwangen.de
†Centre for Security, Communications and Network Research - University of Plymouth
Plymouth PL4 8AA, United Kingdom
Email: N.Clarke@plymouth.ac.uk

***Abstract*—One characteristic of a cloud computing infrastructure are their frequently changing virtual infrastructure. New Virtual Machines (VMs) get deployed, existing VMs migrate to a different host or network segment and VMs vanish since they get deleted by their user. Classic incidence monitoring mechanisms are not flexible enough to cope with cloud specific characteristics such as frequent infrastructure changes. In this paper we present a prototype demonstration of the Security Audit as a Service (SAaaS) architecture, a cloud audit system which aims to increase trust in cloud infrastructures by introducing more transparency to user and cloud provider on what is happening in the cloud. Especially in the event of a changing infrastructure the demonstration shows, how autonomous agents detect this change, automatically re-evaluate the security status of the cloud and inform the user through an audit report.**

***Keywords*-cloud computing; cloud audit; cloud security**

I. Introduction

Cloud Computing is more than just a current hype, and that's not just because it is the main topic at the worlds largest computer exhibition Cebit for the third year in a row. In a comparative study between 700 medium and large enterprises in 2011, one third explained that they already use cloud services for 25% of their systems in the back office infrastructure. Cloud Computing is a rapidly growing market, but mainly security reasons hinder a broad industry acceptance. This is reflected in the increasing number of research funding going into projects to enhance cloud computing security and standards, such as the EU's FP7 Objective 1.2 "Internet of Services, Software and Virtualization or Germany's economic development scheme "Trusted Cloud".

One reason while traditional security instruments and best practices are not flexible enough to cope with new challenges of cloud computing [1] is the increasing complexity. Incidents at Amazon Web Services (AWS) in early 2011 show that cloud user must adopt their security, backup and business continuity strategies to the cloud paradigm, or in worst case valuable data get lost [2], [3], [4]. While having a system's image as a backup server which will be started in case of a "hardware" failure is a widely accepted business continuity solution this can fail in cloud computing, as AWS customer experienced [5]. After an partial outage of AWS's electronic block storage system (EBS) a huge amount of requests to mirror not available cloud instance to a different storage unit caused a mirroring storm and exhausted the available capacity of the EBS backplane [5]. As a result the EBS system became unable to service "create volume" API requests, resulting in not working backup strategies of AWS customers. As a result of the outage some EC2 customers permanently lost data, although services were hosted on different EC2 availability zones [3].

From the cloud provider's point of view, running and maintaining a cloud infrastructure is more challenging than a classic data center. The reasons lie in cloud computing's characteristics, mainly its multi-tenant user model. A provider has to prove: compliance to laws, especially data protection laws, compliance to laws of all subcontractors, isolation and adequate segregation of shared computing and storage resources, measures taken for availability, service and data protection, e.g. backups, comprehensive continuity-of-operations plan, measures taken to secure the cloud network environment, e.g. Intrusion Detection Systems, firewalls and logging facilities. To fulfill this need governmental and industry security experts, e.g. the German Federal Office for Information Security (BSI), recommend security audits and certificates as the preferred method of proof.

In Germany the BSI maintains the "IT-Grundschutz Catalogues", a guideline to achieve an appropriate security level for all types of information of an organisation [6]. But they also need to adapt to the new characteristics of cloud computing. A cloud audit needs to consider the point of time when the infrastructure changes and the ability to decide if this change gives rise to a security gap or an infrastructure misuse. Knowledge of the underlying business processes is needed, for example, to decide if an up-scaled cloud service is caused by a higher demand of business requests or by hacker misuse. That's why the BSI published a first guideline for cloud computing service provider (CSP) [7] with recommendation how to secure a cloud infrastructure dependent on the protection level of the assets stored in a cloud.

To adapt IT security audits to cloud computing the Cloud Research Lab at University of Applied Sciences Furtwangen

978-0-7695-4756-5/12 $26.00 © 2012 IEEE
DOI 10.1109/SERVICES.2012.12

IEEE computer society

is participating in a collaborative research project "Security Audit as a Service" funded by the German Ministry for Education and Research (BMBF) to investigate how audits of cloud infrastructure can enhance transparency and therefore trust in cloud environments. The developed "Security Audit as a Service (SAaaS)" prototype presented in this work and the appendant SERVICES CUP 2012 demonstration shows how autonomous agents can react on cloud infrastructure changes and automatically re-validate the security status and compliance of IT-security policies after the infrastructure change. This results in a *Concurrent Cloud Audit*.

In this paper, we first describe related work (Section II), followed by selected cloud audit use cases discussed in Section III. Section IV introduces the Security Audit as a Service (SAaaS) architecture. The concept of using a distributed agent framework is introduced and the SAaaS agent and the SERVICES CUP 2012 demonstration prototype is presented in Section V. Section VI evaluates the advantages of autonomous agents for cloud audits. Section VII concludes the paper and informs about future work.

II. Related Work

First related projects get presented which support the basic idea of increasing security in cloud infrastructures by cloud audits. Second research regarding the usage of autonomous agents to overcome traditional monitoring system limitations is presented.

Li et al. present in [8] a method how cloud storage services can benefit from a trusted third party audit (TPA). They introduce issues and solutions for the application of a TPA, such as protection for data integrity, support of dynamic data, access control batch audits and minimized audit costs. While the paper presents valuable ideas for auditing a Storage as a Service cloud model, the presented SAaaS architecture focuses on a wider audit of a cloud infrastructure and on the challenge that an infrastructure change can lead to a new rating if a cloud offer is still considered secure. The presented TPA ideas could be used by the SAaaS architecture to secure data storage.

Wang et al. present in [9] a system to audit integrity and security of public data cloud storage. Their solution allows a third party auditor to be able to efficiently audit the cloud data storage without demanding a local copy of data. Public key based homomorphic authentication is combined with random masking to get a privacy-preserving public cloud data auditing system.

A "Dynamic Audit Services for Outsourced Storages in Clouds" is presented by Zhu et al in [10]. The approach uses fragment structures, random sampling and index-hash tables, supporting provable updates to outsourced data and timely anomaly detection.

Wei et al. present in [11] a VM image management system, which addresses the issue of security patches for VM images. By tracking image access and image provenance a prototype of an image scanner is presented which evaluates the software and its version installed on a VM. If not installed security updates for a certain software exist actions can be defined like warning the user or prevent the start of VMs based on that image. The presented SAaaS architecture will also include a similar image audit methodology. Furthermore a product independent image parser is developed to deliver missing patch information of VM images. This part is under current development and will be published independently and is therefore not described in more detail in this paper.

In [12] and [1] the authors recommend the usage of TPM/TCPA crypt chips to achieve a trusted computing base by a secure OS installation. The SAaaS architecture can utilize all of the above introduced techniques to establish a trusted computing base for cloud environments and extends them to provide a cross customer trust.

To improve the shortcomings of traditional Intrusion Detection & Prevention Systems (IDS, IPS) especially in frequently changing environments the advantages of using distributed, autonomous agents is frequently discussed and demonstrated in [13], [14], [15], [16].

Zamboni et al. present in [17] how IDS could be enhanced by using autonomous agents. They show advantages of using agents in regards to scalability and the detection of system overlapping security incidents. In contrast to our SAaaS architecture their research is focusing on the detection of intrusions into a relatively closed environment whereas our work applies to an open (cloud) environment where incidents like abuse of resources needs to be detected.

Chirumamilla et al. show in [14] that agents can enhance the security of wireless networks they don't really benefit from typical agent characteristics like deployment on demand as the SAaaS agents presented in this paper.

The SAaaS prototype and demo presented in this paper is a successor of the constructional and design work presented in prior papers, such as [18] and [19]. It enhances the presented work by new target scenarios for cloud audits and further implementation of the SAaaS prototype.

III. Targeted Cloud Audit Use Cases

While cloud environments cause new challenges to traditional IT security audits due to their characteristics, they also enable new business cases to perform security audits on a regular basis. This section discusses the following possible use cases for cloud audits: A. *Audit of cloud IT for cloud customer* and B. *Audit of cloud IT for cloud provider*.

A. Audit of cloud IT for cloud customers

Assumed that a cloud customer already uses a cloud offer and runs some instances (VMs) in a cloud. Due to cloud computing's characteristics and their resulting challenges [20], he is facing the following problems: Missing monitoring of cloud instances, data security due to unknown

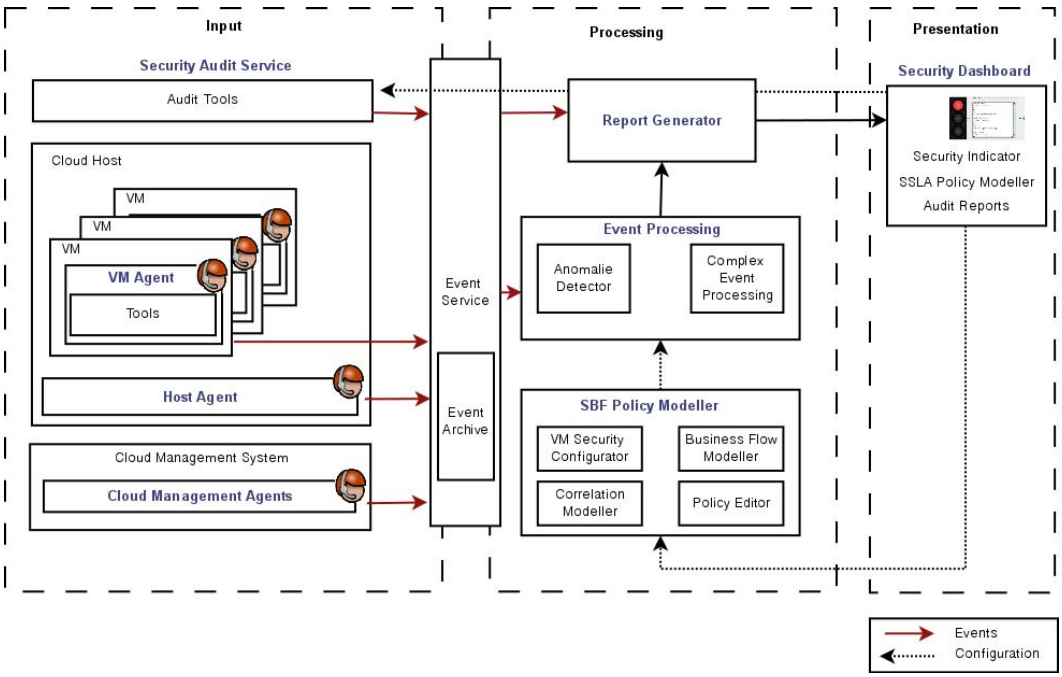

Figure 1. SAaaS event processing flow

data location and threads due shared technology, missing auditability of cloud provider due to missing transparency, and loss of overview due to frequent infrastructure changes (VM start/stop). In the traditional data center scenario, the server landscape does not change often and especially SMEs administrators know their systems by heart. In cloud computing infrastructure may change because of the scalability of cloud resources, resulting in a changing number of active cloud instances to fulfill the service demands. To monitor this changing infrastructure and detect security incidences each cloud instance and the corresponding cloud infrastructure components, e.g. virtual switches, VM hosts, router and switches are monitored. An agent framework is used, where `audit agents` are deployed at all core components of a cloud infrastructure. Each agent is producing events in case of a ominous transaction. The cloud customer is able to define Security Service Level Agreements (SSLAs), that regulate which components should be monitored and how. Furthermore alarm levels describing how the system automatically reacts in the event of a detected security incident. This introduces the following advantages for a cloud customer: better overview of all customer associated instances, possibly created from multiple accounts, transparency about cloud instances' security state and better transparency about provider's administrative access.

B. Audit of cloud IT for cloud service provider

Traditional IT security audits or penetration tests need to be adapted to a cloud's specific attributes, as described in the introduction. Due to the frequently changing infrastructure, the possibility, that possible misusers of cloud resources are already within the cloud's network, and the usage of the resources on demand, traditional audit methods and tools have weaknesses and therefore a concurrent security audit is needed. This is achieved by a monitoring system built on audit agents can provide the following advantages for a cloud provider: detection of attacks against the cloud management system, evaluation of cloud usage behaviour to detect misuse of cloud resources, support of IT forensic investigations in case of successful attacks, reporting of security state of cloud infrastructure over time, monitoring of law compliance. Furthermore an interface to third party security provider for an external audit is provided.

IV. SAaaS Architecture

This section briefly introduces the Security Audit as a Service (SAaaS) infrastructure. The description focuses on the main parts which are used in the SERVICES CUP 2012 demonstration. A more detailed description can be found in our previous paper [21].

Figure 1 gives a high level overview how events are generated, preprocessed, combined and forwarded within the SAaaS architecture. It can be divided into three logical layers: `Input`, `Processing` and `Presentation` layer.

`Input Layer`: The SAaaS architecture gets its input information from distributed agents which are positioned at key points of the cloud's infrastructure. Possible key

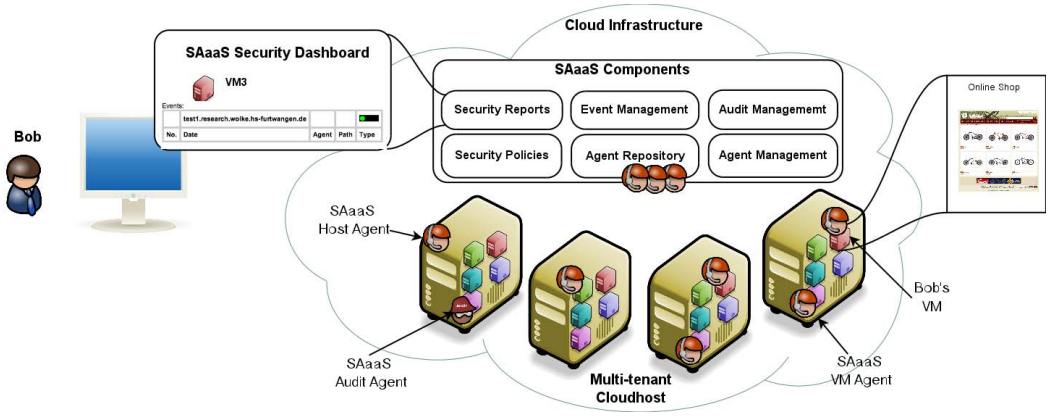

Figure 2. SAaaS components of the SERVICES CUP 2012 demonstration

points are: running VMs of cloud users, the VM hosting systems, data storage, network transition points (e.g. virtual switches), hardware switches, firewalls, and especially the cloud management system. A *VM agent* integrates several monitor and policy enforcing tools. Therefore it loads necessary VM agent plugins to interact with stand-alone *tools* like process monitor, intrusion detection system or anti virus scanner.

`Processing Layer`: Each SAaaS agent receives security policies from the security business flow policy modeler (SBF). Through security policies each agent gets a rule set (its intelligence) specifying actions in case of a specific detected occurrence, such as creation of a new VM. Currently agents are predefined by a manually generated template. The development of a generic API is up to future work. For the SERVICES CUP 2012 a hard coded SBF template is used to validate the compliance of an IT security policy defining requirements for the configuration of a web server VM. The *Report Generator* conditions events, corresponding security status as well as audit report results in a human friendly presentation.

`Presentation Layer`: As a single interaction point to cloud users the *Security Dashboard* provides usage profiles, trends, anomalies and cloud instances' security status (e.g., patch level). Information are organized in different granular hierarchies depending on the information detail necessary. At the highest level a simple three colour indicator informs about a users cloud services overall status.

Why using an agent framework?

To be able to react on frequent infrastructural changes, concurrent audits could be used to re-validate a security status after a change happened. This process has to be lightweight to perform even in large installations, like a cloud computing environment. Our research has shown [21] that using an agent framework is viable to achieve this task. The main advantages of agents are:

Reduction of events: Especially the frequently changing infrastructure is a big challenge to evaluate a security status of an cloud instance and its business case in a cloud environment. Therefore it is important to have a high number of sensors capturing simple events. Simple events can be preprocessed and abstracted to complex events, reducing the possibility "of event storms".

Fast adaption: Combined with knowledge about business process flows it will be possible to detect security incidents in a frequently changing infrastructure while keeping the network load low. The usage of autonomous acting agents delivers this possibility.

Flexibility: Agents can be added, reconfigured or removed during runtime without touching other components. Thus, the amount of monitoring entities (e.g., network connections of a VM, running processes, storage access, etc.) of a cloud instance can be changed without restarting e.g. a whole monitoring system, like a host based intrusion detection system. If underlying business processes are taken into account using lightweight agents can save computing resources.

Agent code example

To give a short introductive example of a SAaaS agent implementation, Listing 1 shows a code snipped of a Manager-Agent. It contains a cyclic behaviour which checks the event database for an infrastructure change event (line 1 - 12). If an event is found (line 9 - 15), a request for deploying a new AuditAgent to the new host where this event was triggered from is made. As a consequence, the AgentRolloutManager configures the newly created AuditAgent and deploys it to the target host (line 17 - 27).

```
1  //Create new hashmap with fqdn pairs of each event
2  Map <String,String> hosts = Collections.synchronizedMap
       (new HashMap<String,String>());
3
4  //Get DebugAgentEvents for agent config from database
5  List<DebugAgentEvent> debugEvents =
       getNotCheckedDebugEvents("config");
```

```
6
7  //Go through DebugAgentEvent list
8  for (DebugAgentEvent event : debugEvents) {
9     if (!hosts.containsKey(event.getLocation())) {
10       hosts.put(event.getLocation(),
11          event.getHostname());
12    }
13    //set this event to be checked in the database
14    setCheckedDebugEvent(event);
15 }
16
17 // Deploy new AuditAgent
18 for(Map.Entry<String,String> e : hosts.entrySet()) {
19    RolloutInfo rolloutInfo = new RolloutInfo();
20    //add the infos for the host to check
21    rolloutInfo.setHostname(e.getValue());
22    rolloutInfo.setFqdn(e.getKey());
23    rolloutInfo.setVmType("audit");
24
25    //Send message to RolloutManager
26    sendmsg.sendMessage(agentRolloutManager, rolloutInfo
         , ACLMessage.REQUEST);
27 }
```

Listing 1. Cyclic Behaviour of the ManagerAgent label

V. SERVICES CUP 2012 DEMO

This section introduces the Services Cup 2012 SAaaS demonstration structure and procedure. The Cloud Computing Infrastructure and Applications (CloudIA) cloud at the University of Applied Sciences Furtwangen is providing the Infrastructure as a Service. One service of the CloudIA environment is "Security Audit as a Service" (SAaaS), that allows the cloud customer to create IT security policies which apply for the cloud instances. Compliance of the security policies are validated through cloud by concurrent security audits. These regularly performed security audits verify the security state of the cloud on a scheduled basis, or if a change to the cloud infrastructure happened, to verify it's security state. First the *Demo Scenario* is described, followed by the *Java Agent DEvelopment Framework* and the *Lab Setup* description. The technical background for each step of the demo is given in *Demo Step by Step*.

A. Demo Scenario

To show how concurrent cloud audits can be achieved, the following demonstration for the Services Cup 2012 has been developed:

Bob, a web administrator of a company, runs a VM containing a web server hosting an online shop. At the beginning of the demo the security status of Bob's VMs is okay which corresponds to an audit result security status of *green*. During the demo Bob will deploy a new web server VM with a web server configuration which violates Bob's IT security policies (e.g. web server must be configured for SSL communication usage). The SAaaS architecture will detect this through concurrent security audits and sets the security status to *red*.

Company wide IT-Security policies are stored in a machine readable format for the SAaaS audit agents called Security Service Level Agreements (SSLAs). Every time a new web server VM gets created a lightweight SAaaS agent gets configured with the appropriate SSLA and deployed to the VM. Periodically an audit agent is deployed to each VM. The agent checks the security of the VM and summarizes the results in an audit report. The SAaaS security dashboard informs the cloud user about the security status and events on his cloud instances. A simple security indicator, depicted as a traffic light, informs quickly about the security state of a cloud instance.

B. Java Agent DEvelopment Framework (JADE)

The core of the SAaaS architecture is build upon the Java Agent DEvelopment Framework (JADE) which is creating, hosting and running the agents for their specific tasks. JADE is a multi-agent system compliant to the Foundation for Intelligent Physical Agents (FIPA) specifications and implements all of the mandatory components, like naming service, yellow-page service, message transport and parsing service. The communication is done via messages represented in the FIPA-ACL language. One main feature of JADE is the implementation in Java and the cross-platform compatibility of it. Each JADE platform is hosting at least one lightweight main container with three default agents and can hosting zero or more sub-containers. The default agents are the Agent Management System (AMS), Directory Facility (DF) and the Remote GUI Agent (RMA). Newly created agents can run on a main container or on a new created sub-container. Therefore in a SAaaS enabled cloud environment, every VM contains an agent container running an AMS, DF and a RMA agent.

C. Lab Setup

For this demo the cloud environment Cloud Computing Infrastructure and Applications (CloudIA) at the Cloud Research Lab of the University of Applied Science Furtwangen has been used. As a cloud management system CloudIA is built on OpenNebula 3.0 and KVM 0.12.5 as virtualization technology. A main management VM is running a JADE 4.1.1 platform which contains a Manager Agent to coordinate events and audit reports and an Agent Rollout Manager to deploy specific agents to the newly created or altered VMs. To ensure the IT security policies at VMs, on each VM a JADE 4.1.1 platform is running to be able to receive agents and execute them. Each VM is also running an Event Aggregator Agent which is receiving messages from single specific agents on it's platform to preprocess and aggregate them, before sending abstracted events to the Manager Agent. An example is an Inotfiy Agent which watches config files for changes and send out events to the Event Aggregator Agent, in case a file was altered. While this normally corresponds in at least three simple events (file open, file modified, file closed) this can then be correlated into one message: file changed.

At the moment JADE only supports intra platform mobility for the mobility of agents between containers on the same

platform. To be able to move agents between platforms (inter platform mobility) and therefore between VMs, the JADE Inter-Platform Mobility Service (jipms) [22] is used. This add-on adds a new agent called Agent Mobility Manager (AMM) to the JADE main container which is responsible for the mobility of agents between platforms.

All demo actions sent to the JADE platforms are executed via a servlet, using a Jade Gateway for communication between JADE and Non-JADE code. This servlet is running on a Tomcat 6.0.26 application server. The web-based graphical user interface (GUI) of this demo to provide interaction with the servlet is written in PHP.

D. Demo Step by Step

The demo can be reached via browser at: http://saaasgui.research.wolke.hs-furtwangen.de:8080/. Demo user credentials are: username: servicescup2012, password: SAaaSatServicesCup2012. Next it follows a detailed description of the demo, which is is divided into several steps. The starting point is a running VM (Bob's) with a correctly configured web server.

Step 1: In the first step, depicted in Figure 3 it is possible to check if the VM is running and the latest audit report can be shown. Clicking on a button sends a get request to the servlet. The servlet processes the request and performs the action. A servlet action serves as an interface between the SAaaS web interface and the SAaaSJadeGatewayAgent which passes the corresponding actions to the other JADE platforms. Communication with the JADE platform (agents) is always performed through the JadeGateway.

The first button of the demo "Get running VMS" calls the servlet action "getvms". This action sets up a between the demo webgui and OpenNebula and returns a list of currently running VMs of a user. The second button "Get last report" invokes the action "auditreport" of the servlet. The servlet processes it and requests the audit report from the manager agent on the JADE main platform. This audit report shows a green security indicator because the last audit of the VM was OK and nothing wrong was found.

Step 2: In the next demo step three different actions are to be performed as shown in Figure 4. First, a new VM is to be created, second, its reachability is to be checked and third, the minimum necessary SAaaS agents are to be deployed to the VM. the third action is happening automatically. during the creation process a VM template is initialized. The template is sent to Open Nebula which creates a second web server VM. The checking of the reachability is done by requesting information from a running agent on the created VM. If it is reachable the action for deploying agents is invoked. The Rollout Agent gets the information to deploys an Event Aggregator Agent and an Inotify Agent to the newly created VM.

Step 3: Step three of the demo are showing the modification and in the next step the notification process. After

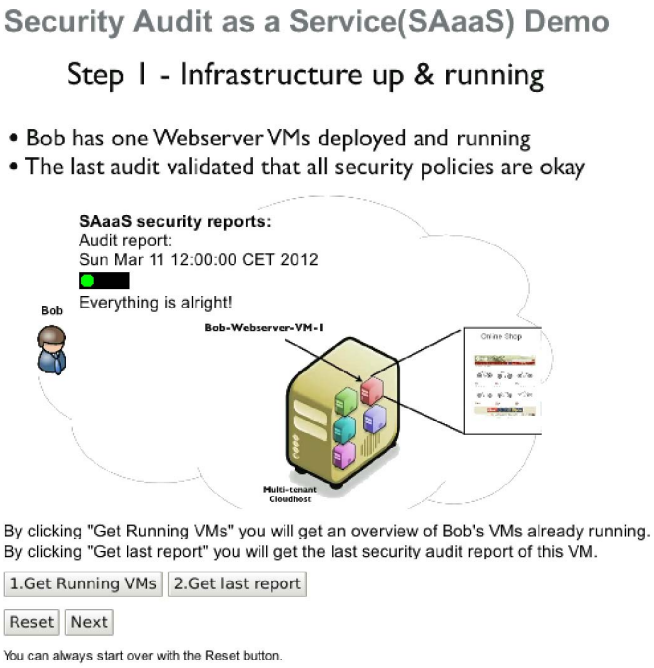

Figure 3. SAaaS demonstration step 1

the configuration has been changed, the Rollout Agent on the JADE main platform deploys the Config Change Agent which alters the web server configuration. The running Inotify Agent notices the changes of the web server configuration and sends an event to the Event Aggregator Agent.

Step 4: Periodically the Event Aggregator Agent sends new events to the Manager Agent. When the Manager Agent receives a new event, it initiates the deployment of an Audit Agent. Again the Rollout Agent deploys an agent (Config Audit Agent) to the VM. This agent detects the invalid web server configuration and sends his report to the Manager Agent. These events are periodically queried by the GUI frontend and displayed on the right.

Step 5: In step five the new audit report gets displayed. The servlet requests the audit report from the Manager Agent through the JADE Gateway and displays it on the right. The red security indicator tells that something is wrong with the VM and a small text is displayed for more information.

VI. Evaluation

Using an agent-based audit system in cloud environments is of advantage because of the adaptability to many different hosts hosting many different services. Small agents, programmed for one single task are using very few resources and when finished with the task the agent can be delete easily. Audit agents are moved to a VM to e.g. check the configuration files from the included Apache web server for entries which are violating against defined IT-security policies. These audits can be made dynamically on a specific

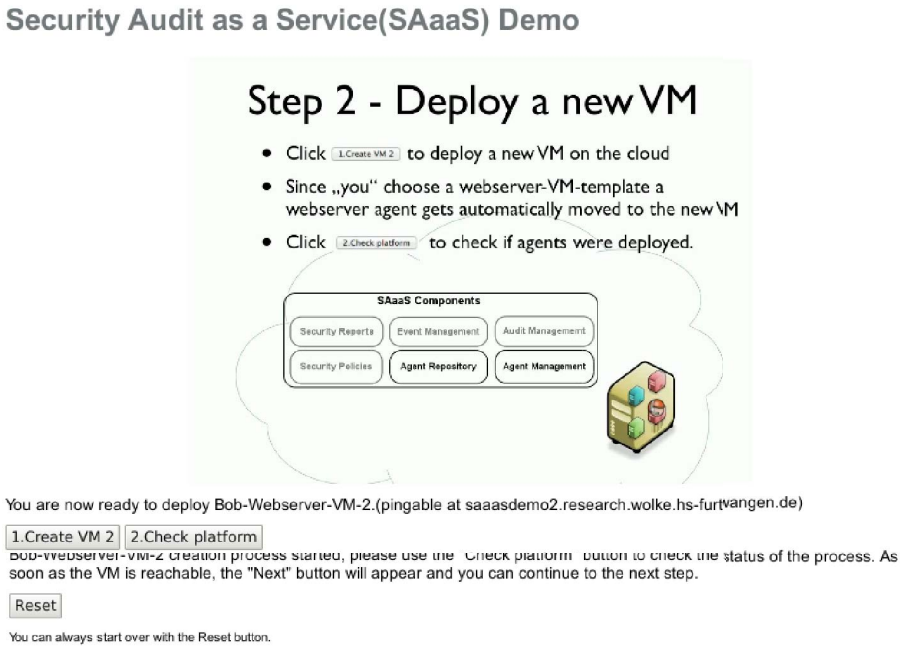

Figure 4. SAaaS demonstration step 2

VM or a given range of VMs to check the security status and are done by request from a user or as a reaction to an event triggered by an agent on the watched VM. If such a security problem is detected at one VM an audit on similar types of VMs can be done.

When a simple event is produced it first gets processed by the agent, which is initiating the event. Afterwards this agent informs all other agents which are also involved in the current business case (agent group). This is important to reduce the overall messages sent to the cloud event processing system especially in large cloud computing environments. Imagine a high demand on a web server, which gets detected by the web server agent. All web server requests will result in database queries which can result in a high load of events produced at the database VM's database agent. By informing the business flow participating agents (web server agent $\rightarrow$ database agent) with an abstract message.

Furthermore, abstracted business flow events can be distributed to a cloud infrastructure monitoring agent. This could be a started web shop request from a specific src IP. A more abstracted event gets sent to the cloud event processing system to detect (possible) user overlapping security incidences. This could be the number of incomplete web shop transactions originated by the same source IP at other web shop (of other customers) as well. Thus a detection of a Denial of Service attack can be done.

A security analysis will investigate the attack detection of the targeted cloud security issues including false positive rates. Further investigation will be done to clarify the following questions: How easy would it be to disable or deceit the security audit system by a malicious user? How well isolated are the security audit systems that are set up for different tenants? What would be possible consequences of maliciously disabling or hijacking an agent?

VII. Conclusion

Concurrent IT security audits can be useful to increase transparency and trust in cloud computing environments. It is shown that they need to respect the specific characteristics of a cloud infrastructure, like frequent infrastructure changes. The target use case scenarios of the presented SAaaS are discussed and a brief description of the SAaaS architecture components was given. To show the possibilities of an agent based cloud audit system the Services Cup 2012 SAaaS demonstration was presented and explained in detail. The demonstration explains the concept of concurrent audits which re-validate the security status of a cloud users cloud instances by performing automatic security audits in case of an infrastructure change, Concluding the advantages of the presented demonstration and using agents was shown.

Following the development of the SAaaS prototype, the sub and complete system will be evaluated due to its scalability. Experiments with increasing VM and agent count (10,50,100,...) will be developed to validate that the SAaaS approach is scalable.

Acknowledgment

This research is supported by the German Federal Ministry of Education and Research (BMBF) through the research grant number 01BY1116.

REFERENCES

[1] L. Vaquero, L. Rodero-Merino, and D. Moran, "Locking the sky: a survey on iaas cloud security," *Computing*, vol. 91, pp. 93–118, 2011.

[2] Business Insider, "Inside Amazon's Cloud Disaster," http://www.businessinsider.com/amazon-outage-enters-its-second-day-lots-of-sites-still-down-2011-4, 04 2011.

[3] B. Insider, "Amazon's Cloud Crash Disaster Permanently Destroyed Many Customers' Data," http://www.businessinsider.com/amazon-lost-data-2011-4, 04 2011.

[4] T. Hoff. (2011, 04) The big list of articles on the amazon outage. http://highscalability.com/blog/2011/4/25/the-big-list-of-articles-on-the-amazon-outage.html.

[5] A. W. Services. (2011, 04) Summary of the amazon ec2 and amazon rds service disruption in the us east region. http://aws.amazon.com/de/message/65648/.

[6] Federal Office for Information Security (BSI). IT-Grundschutz Catalogues. https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html.

[7] F. O. for Information Security (BSI), "Eckpunktepapier Sicherheitsempfehlungen fr Cloud Computing Anbieter," Tech. Rep., 2011.

[8] L. Li, L. Xu, J. Li, and C. Zhang, "Study on the third-party audit in cloud storage service," in *Cloud and Service Computing (CSC), 2011 International Conference on*, dec. 2011, pp. 220 –227.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[10] Y. Zhu, G. Ahn, H. Hu, S. Yau, H. An, and S. Chen, "Dynamic Audit Services for Outsourced Storages in Clouds," *Services Computing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[11] J. Wei, X. Zhang, G. Ammons, V. Bala, and P. Ning, "Managing security of virtual machine images in a cloud environment," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, ser. CCSW '09. New York, NY, USA: ACM, 2009, pp. 91–96. [Online]. Available: http://doi.acm.org/10.1145/1655008.1655021

[12] Andreas Antonopoulos, "Securing Virtualized Infrastructure: From Static Security to Virtual Shields," Nemertes Research, Tech. Rep., 2007.

[13] C.-L. Lui, T.-C. Fu, and T.-Y. Cheung, "Agent-based network intrusion detection system using data mining approaches," in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 1, july 2005, pp. 131 – 136 vol.1.

[14] M. Chirumamilla and B. Ramamurthy, "Agent based intrusion detection and response system for wireless lans," in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 1, May 2003, pp. 492 – 496.

[15] D. Dasgupta, F. Gonzalez, K. Yallapu, J. Gomez, and R. Yarramsettii, "Cids: An agent-based intrusion detection system," *Computers & Security*, vol. 24, no. 5, pp. 387 – 398, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404805000179

[16] C. Krgel and M. A. Based, "Sparta - a mobile agent based intrusion detection system," in *Proceedings of the IFIP conference on network security*. Kluwer Academic Publishers, 2001.

[17] J. Balasubramaniyan, J. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, "An architecture for intrusion detection using autonomous agents," in *Computer Security Applications Conference, 1998, Proceedings., 14th Annual*, dec 1998, pp. 13 –24.

[18] F. Dölitzscher, C. Reich, and A. Sulistio, "Designing cloud services adhering to government privacy laws," in *CIT*, 2010, pp. 930–935.

[19] F. Doelitzscher, C. Reich, M. Knahl, and N. Clarke, "Incident detection for cloud environments," in *Proceedings*, no. 978-1-61208-174-8, 2011, pp. 100–105, iSBN: 978-1-61208-174-8. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=emerging_2011_5_30_40134

[20] Cloud Security Alliance, "Top Threats to Cloud Computing V1.0," 2010. [Online]. Available: https://cloudsecurityalliance.org/topthreats.html

[21] F. Dölitzscher, C. Reich, M. Knahl, and N. Clarke, "An autonomous agent based incident detection system for cloud environments," in *CloudCom*, 2011, pp. 197–204.

[22] J. Cucurull, R. Mart, G. Navarro-Arribas, S. Robles, B. Overeinder, and J. Borrell, "Agent mobility architecture based on ieee-fipa standards," *Computer Communications*, vol. 32, no. 4, pp. 712 – 729, 2009.