Towards a Security SLA-based Cloud Monitoring Service

Dana Petcu and Ciprian Crăciun
Computer Science Department, West University of Timişoara, Timişoara, Romania

Keywords: Cloud Monitoring, Security Monitoring, SLA Management.

Abstract: Following the community concerns related to security and trust in cloud services, services level agreements (SLAs) are nowadays revised to include security requirements. In order to speedup their take-up by the service providers and consumers, security SLAs monitoring at run-time should be ensured. Several tools for SLA management are available, but most of them are dealing with performance parameters, and not referring to security. Other tools are available for cloud security monitoring, but not currently related or mapped to security SLAs. Aiming to design and develop a security SLA-based cloud monitoring service, which can be deployed or hosted, we identify in this paper the concepts, mechanism and available tools which can lead to a proper design of such a service, as well as the main barriers to overcome.

1 INTRODUCTION

The monitoring process is a key element to be enhanced for upgrading the quality level of the current cloud services. The monitoring of the cloud and of its service level agreements (SLAs) in simple terms like workload, performance or availability, offers both to the providers and the consumers the information necessary to implement mechanisms to prevent or recover from agreement violations. However there are only a few SLA management systems which are hosted or deployable in clouds.

Security monitoring is less developed than operational performance monitoring in cloud environments. Moreover, it is laging behind other security features at cloud providers. Security obligations associated with a service should be specified at request in a SLA (in security SLA, or, shortly, Sec-SLA). The absence of security aspects in the currently used SLAs, combined with the lack of methods for making objective comparisons between different service offerings, makes it impossible for providers to offer trustworthy services to their customers (Bernsmed et al., 2011). Moreover, according (Wagner et al., 2012), cloud services provider contracts will not provide detailed and substantive security SLAs before 2016.

In order to address these problems, we are interested in developing an open-source SLA-based Cloud security monitoring system that can act as monitoring-as-a-service or can be deployed in conjunction with the open-source mOSAIC PaaS (Petcu et al.,) which offers a certain degree of portability of applications consuming IaaSs. An incipient form of SLA-based monitoring system with no security parameters was designed, prototyped and reported in (Rak et al., 2011). The role of a new SLA-based cloud security monitoring service, in the SPECS framework based on mOSAIC PaaS, under design and development, was exposed in (Rak et al., 2013): a Sec-SLA that is negotiated will be monitored for compliance, and alerts will be generated in case of security changes or in case of Sec-SLA violations, leading to its enforcement. In this paper we discuss the availability or lack of adequate concepts, mechanisms and tools to be reused when such a service is designed. The next section identifies the state-of-art, while its following section evaluates the appropriateness of the existing solutions for the new service.

2 RELATED WORK

We have recently collected data and reports about the current tools and prototypes that are available for SLA monitoring or security monitoring in clouds. Tables 1-2 are pointing towards the most significant ones (not exhaustive lists). Here the tools and frameworks are organized in three categories: open-source, commercial or research prototypes.

Surprisingly, there is no report until now of any effort for developing a Sec-SLA based Cloud monitoring service. In what follows we point to some of

Petcu D. and Crăciun C..
Towards a Security SLA-based Cloud Monitoring Service.
DOI: 10.5220/0004957305980603
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 598-603
ISBN: 978-989-758-019-2

Table 1: Open-source and Commercial Cloud monitoring tools which are SLA-based or security-oriented.

Acronym	Short description
	Open-source SLA-oriented Cloud monitor tools
CloudCompass[1]	SLA-aware PaaS featuring an extension of the SLA specification WS-Agreement for cloud. The monitor module performs dynamic assessment of the QoS rules from active SLAs.
Sandpiper[2]	Automates the process of monitoring, detecting hotspots and reconfiguring VMs whenever necessary. The monitoring system uses thresholds to check whether SLAs can be violated.
SLA@SOI[3]	SLA management platform that includes at monitoring layer EVEREST, a general-purpose engine for monitoring the behavioural and quality properties of distributed systems based on events
	Open-source Cloud security monitor tools
FBCrypt[4]	Prevents information leakage; it encrypts the I/O between a client and a user VM using a virtual machine monitor -VMM: intercepts reads of virtual devices by a user VM and decrypts inputs.
Snorby[5]	Application for network/host security monitoring; integrates with intrusion detection systs (IDS)
	Commercial SLA-oriented Cloud monitor tools
NMS[6]	'NimSoft Monitoring Solution' is a unified monitoring dashboard able to monitor data centers of both private and public Clouds; it can be used for monitoring SLAs.
Site24x7[7]	Online website monitoring service continuously checking the availability of web-sites/apps; it includes application performance monitoring, SLA management, uptime reporting.
	Commercial Cloud security monitor tools
CipherCloud[8]	Service and virtual appliance delivering a set of protection controls including encryption, tokenization, activity monitoring, data loss prevention, malware detection.
CloudFlare[9]	Protects websites (optimize delivery, blocks threats, limit abusive bots and crawlers): after the enrolement of the website in the community, its web traffic is re-routed.
CloudPassage[10]	Automate security and compliance: integrated into platforms like OpenStack or by providers like AWS, it features a cloud-agnostic architecture, continuous security monitoring/control, and REST APIs for integration with automation tools like Puppet and security systs like Splunk
MARS[11]	Cisco's 'Monitoring, Analysis & Response System' designed to monitor logs and threats
SPAE[12]	'Security Performance Availability Engine' is a network monitoring tool supporting a various protocols and using SNMP; enables consumers to monitor networked Cloud resources.
Splunk' Storm[13]	Cloud-based service for analyzing machine data generated by web sites, applications, servers, networks, mobile device.
ThreatStack[14]	Deployable, profiling for normal behaviour, real-time monitoring: user loggings, network connections, data send to ThreatStack analyzer servers, firewall policies, forensics, audit alerts

the issues that have led to this current status, with accent on complexity and novelty.

Cloud monitoring is done at high or low level (Aceto et al., 2013). A high-level monitoring is related to information on the status of the virtual platform, collected at the middleware, application and user layers by service providers or consumers through platforms and services operated by themselves or by third parties. A low-level monitoring is related to information collected by the service provider and usually not exposed to the consumer, and is concerned with the status of the physical infrastructure.

The security monitoring is falling mostly in the category of high-level monitoring. For low-level monitoring specific utilities for collecting information about security are referring to software vulnerabilities or bugs (at OS and middleware layer), IDS or firewalls (at network layer), authentication systems or surveillance (at facility layer), workload, voltage or temperature, memory or CPU (at hardware level).

The definition of a security that can be quantifiable and can be expressed in a service level is a very complex task (de Chaves et al., 2010). In particular the definition of the security metric is challenging and

[1] github.com/angarg12/cloudcompaas-common
[2] lass.cs.umass.edu/projects/virtualization/sandpiper
[3] sourceforge.net/apps/trac/sla-at-soi/
[4] ksl.ci.kyutech.ac.jp/oss/fbcrypt
[5] github.com/snorby/snorby
[6] www.nimsoft.com/solutions/cloud-monitoring
[7] www.site24x7.com
[8] www.ciphercloud.com
[9] www.cloudflare.com/overview
[10] cloudpassage.com
[11] www.cisco.com/en/US/products/ps6241/
[12] shalb.com/en/spae/spae_features/
[13] www.splunkstorm.com
[14] www.threatstack.com

Table 2: Research prototypes of Cloud monitoring tools which are SLA-based or security-oriented.

Acronym	Short description
	SLA-oriented Cloud monitoring tools
CASViD	'Cloud application SLA violation detection' aims at SNMP-based monitoring and detecting SLA violations at application layer; it includes tools for resource allocation (Emeakaroha et al., 2012).
LoM2HiS	'Low-level Metrics to High-level SLA monitoring and mapping' monitors resource metrics and maps the metric values to high-level SLA parameter objectives (Emeakaroha et al., 2010).
QoS-MONaaS	'Quality of Service MONitoring as a Service' allows to describe in a formal SLA the key performance indicators of interest and the alerts in case of SLA violation (Adinolfi et al., 2012).
	Cloud security monitoring tools
Aftersight	Can be used to analyze the behavior of a VM; it decouples execution of the VMs from this analysis of the execution; it records non-deterministic events, inputs to a VM (Chow et al., 2008).
CloudSec	Provides active, transparent and real-time security monitoring for multiple concurrent VMs hosted on a cloud platform in an IaaS setting (Ibrahim et al., 2011).
CloudWatcher	Security monitoring as a service automatically detouring network packets to be inspected by pre-installed network security devices (Shin and Gu, 2012).
HyperWall	A hypervisor that is not be able to snoop on, or modify, the data (or code) that is exchanged between the VM and the resource, or on computation done in the VM (Szefer, 2013).
K-Tracer	Dynamically analyzes Windows kernel-level code and extracts malicious behavior from rootkits, based on QEMU virtualization technology (Lanzi et al., 2009).
KVMSec	Extension to the KVM to check the integrity of guest VMs by adding modules in host & guest side: guest OS sends information to the host about the VM status (Lombardi and Di Pietro, 2009).
Lares	Based on two VMs: an untrusted monitored VM and a security VM. The last monitors the first and can see into the state of the monitored VM using an introspection API (Payne et al., 2008).
Livewire	An IDS which uses the VMM to pull the intrusion detection logic out of a monitored VM; the IDS VM runs on the same server as the VM being monitored (Garfinkel and Rosenblum, 2003).
Lycosid	Detects hidden process in VMs comparing guest view with a VMM image (Jones et al., 2008).
MAVMM	A VMM for malware analysis extracts features of the applications running inside a guest OS: execution trace, memory pages, system calls, disk accesses, network (Nguyen et al., 2009).
MISURE	'Monitoring Infrastructure using Streams on an Ultra-scalable, near-Realtime Engine' is a monitoring-as-a-service for data analysis; uses stream processors like S4, Storm (Smit et al., 2013)
NICKLE	A VMM based on memory shadowing: the trusted VMM maintains a shadow physical memory for a running guest VM & performs real-time authentication of the kernel code (Riley et al., 2008).
Overshadow	Protects the privacy and integrity of application data in a guest VM even if the guest OS is compromised (the VMM provides guest physical memory pages accordingly) (Chen et al., 2008).
PoKeR	'Profiler of Kernel Rootkits' profiles four rootkit characteristics: hooking behavior, kernel object modifications, impact on user applications, code injection (Riley et al., 2009).
Revirt	Ensure secure logging and logs information: real-time clock, keyboard, mouse events, user inputs, system calls, enabling admin to replay the execution of VM/ analyze attacks (Dunlap et al., 2002).
Rkprofiler	Sandbox-based malware tracking using QEMU virtualization for Windows (Xuan et al., 2009).
SecMon	Secure introspection framework using a VMM for Windows OS (Wu et al., 2013).
SecVisor	Hypervisor supporting one guest VM to protect it from rootkits (Seshadri et al., 2007).
SIM	'Secure-in-vm monitoring': monitoring code in the VM with monitored code (Sharif et al., 2009).
VMWatcher	An out-of-the-box malware detection mechanism addressing the gap between observed events at the VMM level and guest OS context; ensures strong tamper-resistance (Jiang et al., 2007).
TrustVisor	Hypervisor which protects pieces of application logic to be execute in isolation: the programmer specifies these pieces as well as valid entry and exit points (McCune et al., 2010).

numbers are not appropriate as the security is related to a variety of properties, varying from a service performance to process maturity.

Fortunately, security parameters for a security monitoring framework were defined and classified recently in (Hogben and Dekker, 2012). Beyond the parameter definition, methods and techniques for measuring parameters in practice were defined. Moreover, thresholds were established to indicate when to trigger an event (how to determine the ranges of parameters that would trigger an incident report, or response and remediation based on real-time or regular service

level reports). However security indicators were not provided (a security indicator is an observable characteristic that correlates with a desired security property; the set of feasible indicator values is expected to form a nominal scale).

To overcome this problem, a step forward was made in (Pannetrat et al., 2013) by providing an attribute-based security property vocabulary and by developing security properties in abstract terms and as a properties with a set of defined attributes.

Cloud security monitoring is currently done on-premises, on the monitored infrastructure, or via a SaaS. In the case of monitoring on-premises, a security tool is able to make use of specific APIs as well to collect logs from cloud services. In the second case, of monitoring on monitored IaaS, a security tool is loaded directly into an IaaS (no high bandwidth requirement, possible some high storage costs, but currently there is a lack of a unified view on the approach). In the third case, monitoring data is obtained from the cloud service (if available), and hand it to a managed security service provider.

Looking at the available tools for Cloud security monitoring systems displayed in Tables 1-2, we see that most of them are low level. Their approach is either to take a complete VM as the monitoring granularity, such that they cannot capture the security incidents within individual VMs, or to focus on specific monitoring function that cannot be used for heterogeneous VMs concurrently running on one cloud node (Zou et al., 2013).

The few SLA management systems that are including monitoring features are comprehensive in terms of covering various cloud services. In particular several domain specific languages were developed to describe the monitored properties present in the SLA as well as the alert or SLA violation thresholds (e.g. in SLA@SOI and QoS-MONaaS, the first being re-used in various follow-up research projects). LoM2HiS is the first try to map high-level parameters objectives to low-level metrics. Despite the degree of granularity of those tools, most of their reported use-cases are concerned with performance monitoring. The security parameters have not been taken into consideration by them.

3 POTENTIAL COVERAGE FOR SEC-SLA MONITORING

We assume in what follows that the Sec-SLA to be adopted by the cloud service providers and consumers follows the vocabulary reported in (Pannetrat et al., 2013), including also its security indicators. We are interested in mapping the available mechanisms and tools to the various security properties from this vocabulary and in filling the gaps where this mapping is not possible. As targeting an open-source service, we are referring here only to the open-source (extensible) tools described in the previous section, as well as to other open-source general monitoring tools, like collectl[15], CloudCmp[16], Cloudstone[17], Ganglia[18], Groundwork[19], Hyperic-HQ[20], JasMINe[21], MonALISA[22], Nagios[23], PCMons[24], SIGAR[25], Zabbix[26].

We have build a matrix of coverage visible in Tables 3 and 4. We considered five levels for indicate if a monitor feature is provided by a service provider or a consumer software (0 – not possible; 1 – there are some serious issues with the proposed property; 2 – although technically conceivable it is less likely to be implemented due to cost & effort; 3 – not usually provided, or it is complicated to provide; 4 – usually happens) according to three categories, IaaS (I), P/SaaS (P: provider oriented) and App (A: consumer oriented, i.e. a VM in case of IaaS, or a software service in case of a PaaS/SaaS). S stands for a deployable service, while L refers to a programming library. We then considered in the last two columns the case of a deployable service and a programming library; we refined further the expectation levels in eleven levels (a – possible with the current tools to monitor and enforce; b – possible with the current tools to monitor; c – if tools are build to monitor/enforce; d – if tools are build to monitor; e – if tools are build to monitor but not straightforward; f – if tools are build to monitor/enforce but it is complex issue; g – if tools are build to monitor but it is complex issue; h – if tools are build to enforce but it is complex issue; i – although technically conceivable it is less likely to be implemented due to cost & effort; j – less likely to be implemented; k – not possible).

Such a matrix indicates not only the hot-spots in developing a Sec-SLA monitoring system (the ones near to zero, or near k), but also the probability that a service provider will comply with the Sec-SLA. For example, as '4' indicates 'easy to comply with', the

[15] collectl.sourceforge.net
[16] github.com/angl/cloudcmp
[17] radlab.cs.berkeley.edu/wiki/Projects/Cloudstone
[18] ganglia.sourceforge.net
[19] sourceforge.net/projects/gwmos
[20] sourceforge.net/projects/hyperic-hq
[21] maven.ow2.org/maven2/org/ow2/jasmine
[22] monalisa.caltech.edu/monalisa__Download.htm
[23] nagios.sourceforge.net
[24] code.google.com/p/pcmons
[25] sourceforge.net/projects/sigar
[26] www.zabbix.com/download.php

Table 3: Matrix of coverage for Sec-SLA monitoring by current mechanisms and tools.

Security property	I	P	A	S	L
Software integrity protection	3	3	2	g	i
Software integrity detection	0	3	2	i	h
Malware protection	0	3	3	k	i
Data alteration prevention	2	0	0	k	k
Data alteration detection	4	4	3	d	d
Data access level	4	4	4	c	c
External data exchange confidentiality	3	4	4	c	c
Authentication of data origin	3	3	3	c	c
Network authenticated server access	4	4	4	c	c
Network mutually authenticated channel	3	3	3	c	c
Non repudiation of origin	3	3	3	e	e
Non repudiation of receipt	2	2	2	i	i
Information flow control: blacklist	4	3	4	a	a
Information flow control: whitelist	4	3	4	a	a
% of systems with time synchronization	2	2	2	g	g
User traceability	2	3	4	i	d
Security event storage integrity level	3	3	3	h	h
Tenant isolation level	4	4	4	h	k
Collocation indistinguishability	3	3	2	k	k
Data portability	3	3	3	c	c
Mean time between incidents	4	4	4	b	b
% of timely incident reports	4	4	4	b	b
% of timely incident resolutions	4	4	4	b	b
User authentication & identity assurance level	4	4	4	a	a
Password storage protection level	4	4	4	a	a
% of timely suspension of unused accounts	4	4	4	b	b
Limitation of failed user authentications	4	4	4	a	a
Inactive session blocking	4	4	4	a	a
Limitation parallel active sessions	4	4	4	a	a
Cryptographic brute force resistance	4	4	4	i	i
Key generation quality	4	1	1	k	k
Key access control level	4	4	4	i	i
Cryptographic module protection level	3	3	2	f	f
% of systems with formal risk assessment	3	3	3	j	j
% of systems with tested controls	3	3	3	j	j
Country level anchoring	3	3	3	a	a
Personal data: consultation ability	2	2	2	k	k
Personal data: modification ability	2	2	2	k	k
Personal data: deletion ability	2	2	2	k	k
Personal data: timely access	2	2	2	k	k
Vulnerability exposure level	3	3	3	h	h
% of timely vulnerability corrections	4	4	4	b	b
% of timely vulnerability reports	4	4	4	b	b
Data deletion quality level	4	1	1	k	k
% of timely effective deletions	4	4	4	k	k
Data leakage detection	2	2	2	f	f
Data leakage prevention	2	2	2	f	f
Storage freshness	4	4	4	g	g
Storage retrievability	2	2	2	g	g
% durability	4	4	4	g	g
Authentication feature count	2	3	3	k	k
Tamper evidence	2	2	2	k	k
Tamper resistance	2	2	2	k	k
% of uptime	4	4	4	b	b
% of processed requests	4	4	4	b	b
% of timely recoveries	4	4	4	b	b
Mean time between failure	4	4	4	b	b
Recovery point objective	4	4	4	j	j
Recovery time actual	4	4	4	j	j
Recovery success ratio	4	4	4	j	j

Table 4: Continuation of Table 3.

Security property	I	P	A	S	L
Elasticity reserved capacity	3	3	3	k	k
% of timely provisioning requests	4	4	4	b	b
Allocation limitation	4	4	4	b	b
Denial of service attack resistance	4	4	4	k	k
% of compliant devices	4	4	4	b	b
% of compliant software	4	4	4	b	b
% of timely configuration change notifications	4	4	4	b	b
Configuration change reporting capability	4	4	4	b	b

38 appearances of '4's in 68 security properties means an expectation of 56% for an IaaS provider to easily comply. This percentage can reflect partially a lack of knowledge about some tools availability or a certain subjectivity in defining implementation difficulty. However, such uncertainty can generate a certain deviation from the above percentage, but not leading it to 100%. With this percentange we can explain why Sec-SLA monitoring systems are not yet in place.

4 CONCLUSIONS

Multiple conceptual and technical barriers must be overcome in order to implement a Sec-SLA monitoring service. Some of them were underlined in this paper: lack of acceptance and maturity of the SLA management systems, difficulty of mapping high level security properties to low level monitoring parameters, lack of deployment-layer agnosticism, extra complexity introduced by the virtualization, and so on. The identification of the available concepts, methods and available tools is only the first step for the implementation of the SPECS's Sec-SLA monitoring system in its two intended versions, deployable or hosted service. The first stable version of the open-source code is expected to be available in one year.

ACKNOWLEDGEMENTS

This research is partially supported by the Romanian grant PN-II-ID-PCE-2011-3-0260 (AMICAS), as preliminary study for the grant FP7-ICT-2013-10-610795 (SPECS).

REFERENCES

Aceto, G., Botta, A., De Donato, W., and Pescapè, A. (2013). Survey cloud monitoring: A survey. *Computer Networks*, 57(9):2093–2115.

Adinolfi, O., Cristaldi, R., Coppolino, L., and Romano, L.

(2012). Qos-monaas: A portable architecture for qos monitoring in the cloud. SITIS '12, pages 527–532.

Bernsmed, K., Jaatun, M. G., Meland, P. H., and Undheim, A. (2011). Security slas for federated cloud services. ARES '11, pages 202–209.

Chen, X., Garfinkel, T., Lewis, E. C., Subrahmanyam, P., Waldspurger, C. A., Boneh, D., Dwoskin, J., and Ports, D. R. (2008). Overshadow: A virtualization-based approach to retrofitting protection in commodity operating systems. *SIGOPS Oper. Syst. Rev.*, 42(2):2–13.

Chow, J., Garfinkel, T., and Chen, P. M. (2008). Decoupling dynamic program analysis from execution in virtual environments. ATC '08, pages 1–14.

de Chaves, S., Westphall, C., and Lamin, F. (2010). Sla perspective in security management for cloud computing. ICNS '10, pages 212–217.

Dunlap, G. W., King, S. T., Cinar, S., Basrai, M. A., and Chen, P. M. (2002). Revirt: Enabling intrusion analysis through virtual-machine logging and replay. *SIGOPS Oper. Syst. Rev.*, 36(SI):211–224.

Emeakaroha, V., Brandic, I., Maurer, M., and Dustdar, S. (2010). Low level metrics to high level slas - lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments. HPCS '10, pages 48–54.

Emeakaroha, V., Ferreto, T., Netto, M., Brandic, I., and De Rose, C. (2012). Casvid: Application level monitoring for sla violation detection in clouds. COMPSAC '12, pages 499–508.

Garfinkel, T. and Rosenblum, M. (2003). A virtual machine introspection based architecture for intrusion detection. NDSS'03, pages 191–206.

Hogben, G. and Dekker, M. (2012). Procure secure. a guide to monitoring of security service levels in cloud contracts. Technical report, European Network and Information Security Agency (ENISA).

Ibrahim, A., Hamlyn-Harris, J., Grundy, J., and Almorsy, M. (2011). Cloudsec: A security monitoring appliance for virtual machines in the iaas cloud model. NSS '11, pages 113–120.

Jiang, X., Wang, X., and Xu, D. (2007). Stealthy malware detection through vmm-based "out-of-the-box" semantic view reconstruction. CCS '07, pages 128–138.

Jones, S. T., Arpaci-Dusseau, A. C., and Arpaci-Dusseau, R. H. (2008). Vmm-based hidden process detection and identification using lycosid. VEE '08, pages 91–100.

Lanzi, A., Sharif, M. I., and Lee, W. (2009). K-tracer: A system for extracting kernel malware behavior. NDSS'09.

Lombardi, F. and Di Pietro, R. (2009). Kvmsec: A security extension for linux kernel virtual machines. SAC '09, pages 2029–2034.

McCune, J., Li, Y., Qu, N., Zhou, Z., Datta, A., Gligor, V., and Perrig, A. (2010). Trustvisor: Efficient tcb reduction and attestation. SP '10, pages 143–158.

Nguyen, A., Schear, N., Jung, H., Godiyal, A., King, S., and Nguyen, H. (2009). Mavmm: Lightweight and purpose built vmm for malware analysis. ACSAC '09, pages 441–450.

Pannetrat, A., Hogben, G., Katopodis, S., Spanoudakis, G., and Cazorla, C. S. (2013). D2.1: Security-aware sla specification language and cloud security dependency model. Technical report, Certification infrastrUcture for MUlti-Layer cloUd Services (CUMULUS).

Payne, B., Carbone, M., Sharif, M., and Lee, W. (2008). Lares: An architecture for secure active monitoring using virtualization. SP '08, pages 233–247.

Petcu, D., Di Martino, B., Venticinque, S., Rak, M., Máhr, T., Esnal Lopez, G., Brito, F., Cossu, R., Stopar, M., Šperka, S., and Stankovski, V. Experiences in building a mosaic of clouds. *Journal of Cloud Computing: Advances, Systems and Applications*, 2:12.

Rak, M., Luna, J., Petcu, D., Casola, V., Suri, N., and Villano, U. (2013). Security as a service using an sla-based approach via specs. CloudCom ' 13.

Rak, M., Venticinque, S., Máhr, T., Echevarria, G., and Esnal, G. (2011). Cloud application monitoring: The mosaic approach. CloudCom '11, pages 758–763.

Riley, R., Jiang, X., and Xu, D. (2008). Guest-transparent prevention of kernel rootkits with vmm-based memory shadowing. RAID '08, pages 1–20.

Riley, R., Jiang, X., and Xu, D. (2009). Multi-aspect profiling of kernel rootkit behavior. EuroSys '09, pages 47–60.

Seshadri, A., Luk, M., Qu, N., and Perrig, A. (2007). Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses. SOSP '07, pages 335–350.

Sharif, M. I., Lee, W., Cui, W., and Lanzi, A. (2009). Secure in-vm monitoring using hardware virtualization. CCS '09, pages 477–487.

Shin, S. and Gu, G. (2012). Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?). ICNP '12, pages 1–6.

Smit, M., Simmons, B., and Litoiu, M. (2013). Distributed, application-level monitoring for heterogeneous clouds using stream processing. *Future Generation Computer Systems*, 29(8):2103–2114.

Szefer, J. M. (2013). *Architectures for Secure Cloud Computing Servers*. PhD thesis, University of Princeton.

Wagner, R., Heiser, J., Perkins, E., Nicolett, M., Kavanagh, K. M., Chuvakin, A., and Young, G. (2012). Predicts 2013: Cloud and services security. Technical report, Gartner ID:G00245775.

Wu, X., Gao, Y., Tian, X., Song, Y., Guo, B., Feng, B., and Sun, Y. (2013). Secmon: A secure introspection framework for hardware virtualization. PDP '13, pages 282–286.

Xuan, C., Copeland, J., and Beyah, R. (2009). Toward revealing kernel malware behavior in virtual execution environments. RAID '09, pages 304–325.

Zou, D., Zhang, W., Qiang, W., Xiang, G., Yang, L. T., Jin, H., and Hu, K. (2013). Design and implementation of a trusted monitoring framework for cloud platforms. *Future Generation Computer Systems*, 29(8):2092 – 2102.