Towards security management in the cloud utilizing SECaaS

JAN MÉSZÁROS
University of Economics, Prague
Department of Information Technologies
W. Churchill Sq. 4, 130 67 Prague 3
CZECH REPUBLIC
jan.meszaros@vse.cz

Abstract: Managing of a cloud environment security is a complex task due to amount of present layers and various technological components. This paper proposes a novel security management approach utilising a trusted third-party security services provider as a solution for the cloud security challenging tasks. The trusted third-party acts as the Security-as-a-Service entity providing automation of security management tasks to its tenants. The main idea of this proposal resides in facilitating and ensuring the compliance with security requirements through top-level security policies definitions transformed into low-level configuration and vulnerability controls. Provided outputs are standardised for interoperability purposes and are suitable for subsequent usage such as audit, forensics, monitoring or customer security ensurance.

Key–Words: cloud computing, security, SECaaS, standardization, security configuration management, vulnerability management, SCAP

1 Introduction

Complexity of cloud services technological background has a significant impact on the services' security. Traditional static security controls can not deal with the dynamic nature of cloud computing, new thinking with regard to cloud computing is needed [1]. Multiplicity of cloud stack layers, number of components included in each layer and heterogenity of tools used for securing single part of each layer yields necessity of interoperability among such tools, standardization and certain automation level.

This paper propose a novel security management approach based on a Security-as-a-Service (SECaaS) concept. The proposed service concept itself can be utilised as private and even public service, the service facilitates securing any of infrastructure, platform and software services provided both publicly and privately. The existing standards are incorporated and the principles of security configuration management and vulnerability management are used.

The proposed SECaaS service can be categorised according to Cloud Security Alliance (CSA) [2] as "#5 Security Assessment" and "#7 Security Information and Event Management".

The introductory chapters of this contribution discuss the cloud services security issues, related security management principles and the contemporary relevant standardization efforts. The following chapters presents certain cloud security management approach. This approach can be used for a custom security solution design and implementation. The last chapters brings an overview of related work, concluding chapter and literature references.

The cloud related nomenclature and definitions used in this paper comply with the NIST definition of cloud computing [3].

2 Cloud computing security challenges

Cloud services are built from multiple interconnected components, starting from physical facilities and hardware, abstraction and virtualisation layers and its interfaces, virtualised hardware, operation systems with encapsulated utilities, services and tools, platform components and finally custom software using databases and plenty of other resources.

This enumeration covers examples of IaaS, PaaS and SaaS underlying technical components whose security has to be managed and assured properly regardless of whether the provider directly owns them or rents them from third parties.

If a SaaS provider rents IaaS or PaaS services, he should consider not only his own software security but also security of all third party services. This is an incredible complex task, due to following issues:

- multiplicity of own technical components and third party services,

ISBN: 978-1-61804-134-0

- heterogenity of the manner, how the components and services enables vendor-specific security related information delivery including these issues:
  - syntax disunity,
  - semantics disunity,
  - encapsulating access channel (web application, web service, log files, messaging, ...) disunity.
- lack of interoperability between security tools,
- storage, archiving and indisputableness of security related information,
- adaptation of security related information for audit, forensics and security management purposes,
- adaptation of security related information for customer purposes.

The security related information incorporate configuration status, log message, system alert and the like from any source.

3 The foundations

The security management approach proposed in chapter 4 deals with the present issues utilizing security management principles and standardization described below.

3.1 Relevant security management principles

Configuration management (CM), originating in military engineering, is defined by NIST [4] as ”collection of activities focused on establishing and maintaining the integrity of products and systems, through control of the processes for initializing, changing, and monitoring the configurations of those products and systems.”

On the basis of CM, NIST gives a definition of *Security-Focused Configuration Management* (SecCM) [4]: ”management and control of secure configurations for an information system to enable security and facilitate the management of risk. SecCM builds on the general concepts, processes, and activities of configuration management by attention on the implementation and maintenance of the established security requirements of the organization and information systems.”

Gartner [5] defines *vulnerability management* as ”a process that can be implemented to make IT environments more secure and to improve an organization’s regulatory compliance posture”. The process includes six steps [5]:

- policy definition,
- baseline the environment,
- prioritize mitigation activities,
- shield the environment using security tools,
- mitigate the vulnerability and eliminate the root causes,
- maintain and continually monitor the environment for deviations from policy and to identify new vulnerabilities.

According to Gartner [5], the four main technology categories for automation of vulnerability management process can be used:

- vulnerability assessment,
- security configuration management and policy compliance,
- IT security risk management, and
- security information and event management (SIEM).

3.2 Security related information standardization

There are some standards defined suitable for *security related information* interchange, which is one of the crucial requirements of the proposed approach.

This paper percieves the *security related information* as any security-relevant information originating from a certain system component or human effort.

The standardized specification set suitable for security related information notation is the Security Content Automation Protocol (SCAP). This protocol developed by NIST [6] ”is a suite of specifications that standardize the format and nomenclature by which software flaw and security configuration information is communicated, both to machines and humans.”

SCAP consists of two major elements called *SCAP components specifications* and *SCAP content*. Current SCAP version 1.2 components specifications comprise a suite of 11 open specifications in five categories [7]:

- *languages* providing standard vocabularies and conventions for expressing security policy, technical check mechanisms and assessment results,
- *reporting formats* providing the necessary constructs to express collected information in standardized formats,

ISBN: 978-1-61804-134-0

- *enumerations* define a standard nomenclature and an official dictionary or list of items expressed using that nomenclature,
- *measurement and scoring systems* used for evaluating specific characteristics of a security weakness and generating a relative severity score,
- *integrity protection* helps to preserve the integrity of SCAP content.

SCAP content includes standardized software flaw, security configuration, and platform identification reference data [7].

3.2.1 Configuration checklists

Configuration management can be automated using the *SCAP-expressed* checklists. These checklists use a standardized language *Extensible Configuration Checklist Description Format* (XCCDF), designed for authoring security checklists and for reporting results or evaluating them [7]. The checklists express what checks should be performed using [7]:

- the *Open Vulnerability and Assessment Language* (OVAL) for automated checks and
- the *Open Checklist Interactive Language* (OCIL) for checks that cannot be performed satisfactorily using OVAL.

OVAL and OCIL can include various enumerations [7]:

- *Common Platform Enumeration* (CPE) for platform definitions,
- *Common Configuration Enumeration* (CCE) describing which security settings should be adressed and
- *Common Vulnerabilities and Exposures* (CVE) defining which software flaw should be adressed.

Some SCAP-expressed checklists are publicly available from Red Hat, Novell, Debian, US government agencies and other organisations[1]. Building own custom SCAP-expressed checklists is recommended through customisation of acquired publicly available checklists.

SCAP-expressed checklists are suitable for notation of human-created security configuration requirements and policies.

[1]For details and links visit http://makingsecuritymeasurable.mitre.org/ website.

3.2.2 Event representation

The *Common Event Expression* (CEE) standardizes the representation of event records in logs to achieve interoperability [8]. CEE proposes [8]:

- common, extensible event record syntax,
- common, extensible taxonomy for events,
- common, extensible set of logging recommendations,
- required characteristics for common log transport.

CEE is suitable for notation of computer-created security information, such as log messages and alerts.

4 The cloud security management approach proposal

The fundamental idea of the proposal consists in the assumption that the overall security level of a complex system in the certain point in time depends especially on

- the proper security configuration of each single component, on
- early identification of known vulnerabilities, and on
- proper security events detection and evaluation.

The PDCA[2] method must be repetitively applied to each of the points above, i.e.:

- the accurate set of controls must be defined according to top-level policies,
- the controls must be initially applied to the actual systems,
- the controls must be continually and periodically measured, evaluated, reported and
- the proper actions must be immediately taken.

[2]the Plan-Do-Check-Act management method made popular by Dr. W. Edwards Deming

ISBN: 978-1-61804-134-0

4.1 Requirements

On the grounds of security-focused configuration management and vulnerability management, following requirements for proposed management approach and prospective supporting service (SECaaS) were defined:

- ability to transform top-level policies to low-level component requirements (quality and security measures, configuration requirements, version requirements etc.),
- continually monitor the environment components for deviation from policy and controls,
- ability to identify and to get rid of known vulnerabilities,
- configuration repository and persistent security information storage existence with history recording,
- ability to provide outputs for other security applications (audit, forensics, SIEM, reporting, etc.),
- ability to be a part of risk management framework,
- standardize inputs and outputs to enhance interoperability,
- track relationships between all system components to avoid reciprocal incompatibility or service unavailibility problems being consequent upon security related changes or updates.

These requirements must be fullfilled by certain SECaaS provider according to this proposal. The following chapter defines the requirements for SECaaS provider capabilities.

4.2 SECaaS provider capabilities

The proposed management approach is based on the idea of SECaaS provider that is able to interact with one or more service consumers in the manner ilustrated on figure 1 in appendix.

The both tenant's security staff and tenant's information systems interact securely[3] with the SECaaS provider using the tenant interface built on web based technologies[4]. This interface enables the tenants to:

- define custom or select predefined machine checkable security controls related to top level policies, that can be different for earch tenant,
- provide outputs from continuous monitoring executed by agents (described below) in various forms of detail and structure on the basis of intended purpose (e.g. monitoring centre or tenant's service customer security ensurance),
- secure and trusted storage of all security related information,
- the SECaaS provider should preferably provide encryption, electronic signing and digital timestamping services.

4.2.1 Security policies and controls

The top-level requirements for any cloud service are transformed into detailed technical controls which are internally recorded into the SCAP-expressed checklist format XCCDF, using OVAL and OCIL languages for expressing granular assessment. This checklist can contain electronic signature allowing validating the integrity, origin, and authenticity of documents [9]. The complete reference for this format can be found in [9].

4.2.2 Agents and dispatchers

The agents provided by SECaaS are distributed to tenants and installed into their technical environment which has to be monitored. The agents are operating on particular technical parts of the cloud service's technical environment, they continuously monitor the target environment. The agents communicates over encrypted channel with the agent dispatchers, that are responsible for determining the tasks assigned to each agent. Each agent's tasks originate from the XCCDF configuration checklist. Each agent also assess the controlled technical resources version numbers against external vulnerability information sources provided by the core system. Agents continuously assess the environment components in intervals determined by the agent dispatcher. Apart from the configurations and versions scanning, agents also collects security related log records. Deviations from defined policies are detected and reported by the agent dispatcher. The agents are designed to operate inside all of IaaS, PaaS and SaaS environments.

Messages from the agents are sent to security messaging bus. Security relevant messages are passed to the standardisation engine, which normalizes the syntax and the semantics of all incoming messages to the CEE notation.

[3] Secure communication channel such as HTTPS or VPN must be used.

[4] web application for human interaction, web services for system interaction

ISBN: 978-1-61804-134-0

4.2.3 Outputs

The standardised outputs are provided to the tenant's arbitrary systems, such as monitoring centre, SIEM[5], tools designed for performing audit and forensics or an interface dedicated to cloud customer's security information subscriptions.

4.2.4 Third-parties

The SECaaS provider should use third-party resources to enhance its security services completeness and quality. The provider should automatically gather information from known vulnerability databases, from recommended secure configuration repositories and from the other similar security sources.

The authorities like government or certification authorities may require an access to SECaaS provider interface to ascertain the provider's compliance with some policies or licencing terms or to gather an aggregated security related information for statistical purposes etc.

4.2.5 The core

The core of SECaaS system should at least consist of a pre-defined low-level controls storage that can help tenants to define initial controls, security intelligence store used for evaluation and interpretation of data collected by agents, security related configuration repository, and vulnerabilities repository.

5 Related work

Barrère, Badonnel and Festor [10] are using OVAL language for vulnerability descriptions in the management plane of autonomic networks and systems. They are utilising the Cfengine tool as the autonomic maintenance system that provides support for automating the management of large-scale environments based on high-level policies [10]. Their main contribution resides in integrating OVAL language into Cfengine through translation from OVAL notation into rules interpretable by Cfengine.

Cloud Security Alliance [1] brings the idea of Security-as-a-Service (SECaaS), enumerating additional distinct SECaaS services and their requirements. Cloud Security Alliance research include the CloudTrust Protocol [11] which is a mechanism by which cloud service consumers ask for and receive information about the elements of transparency as applied to cloud service providers employing SCAP protocol.

[5]Security Incident and Event Management

Lang [12] deals with security policy automation employing the model-driven security approach. He presents the proprietary OpenPMF tool implementing application security policy automation.

Banghart [13] introduces the ideas of security standardisation and automation using SCAP protocol.

There are some open-source projects implementing some SCAP components on the SourceForge.net open source application and software directory. For example ovaldi, escapeeditor, scapexec, escapelibrary, xccdfexec, xccdf2pdf and rectracker.

6 Conclusion

This paper proposed a cloud security management approach based on outsourcing of complex security tasks to Security-as-a-Service provider. The approach presented a solution to solve the cloud computing security challenges described in chapter 2. The security-focused configuration management and vulnerability management together with security information standardisation are the fundamental basis of the proposed approach.

Acknowledgements: The research was supported by the Grant Agency of Czech Science Foundation (grant No. P403/11/0574).

References:

[1] Cloud Security Alliance: Security Guidance for Critical Areas of Focus in Cloud Computing v3.0. Cloud Security Alliance, 2011.

[2] Cloud Security Alliance: Secaas: Defined Categories of Service 2011. Cloud Security Alliance, 2011. https://cloudsecurityalliance.org/wp-content/uploads/2011/09/SecaaS_V1_0.pdf

[3] Mell, P., Grance, T.: Special Publication 800-145. The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology. Gaithersburg: NIST, 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[4] Johnson, A. et al.: Special Publication 800-128. Guide for Security-Focused Configuration Management of Information Systems. Gaithersburg: NIST, 2011. http://csrc.nist.gov/publications/nistpubs/800-128/SP800-128.pdf

[5] Williams, A and Nicollet, M: Improve IT Security With Vulnerability Management, Gartner ID Number: G00127481, May 2005

[6] Waltermire, D., Quinn, S., Scarfone, K., Halbardier, A.: Special Publication 800-126 Revision 2. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2. Recommendations of the National Institute of Standards and Technology. Gaithersburg: NIST, 2011. http://csrc.nist.gov/publications/nistpubs/800-126-rev2/SP800-126r2.pdf

[7] Quinn, S., Scarfone, K., Waltermire, D.: Special Publication 800-117 Revision 1 (Draft). Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.2 (Draft). Recommendations of the National Institute of Standards and Technology. Gaithersburg: NIST, 2012. http://csrc.nist.gov/publications/drafts/800-117-R1/Draft-SP800-117-r1.pdf

[8] Fitzgerald, E.: Common Event Expression (CEE) Overview. The CEE Editorial Board, 2010.

[9] Waltermire, D., Schmidt, C., Scarfone, K., Ziring, N.: NIST Interagency Report 7275 Revision 4. Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2. Gaithersburg: NIST, 2012.

[10] Barrère, M., Badonnel, R., Festor, O.: Supporting vulnerability awareness in autonomic networks and systems with OVAL. In Proceedings of the 7th International Conference on Network and Services Management (CNSM '11). International Federation for Information Processing, Laxenburg, Austria, Austria, 37-45.

[11] Knode, R. and Egan, D.: Digital Trust in the Cloud: A Precis for CloudTrust Protocol (V2.0). Computer Sciences Corporation, 2010.

[12] Lang, U.: Model-driven cloud security: Employ cloud application security policy automation to make cloud security better. IBM Corporation, 2011.

[13] Banghart, J.: Security Information Standardization and Automation. In proceedings of the 6th Annual IT Security Automation Conference, 2010.

ISBN: 978-1-61804-134-0

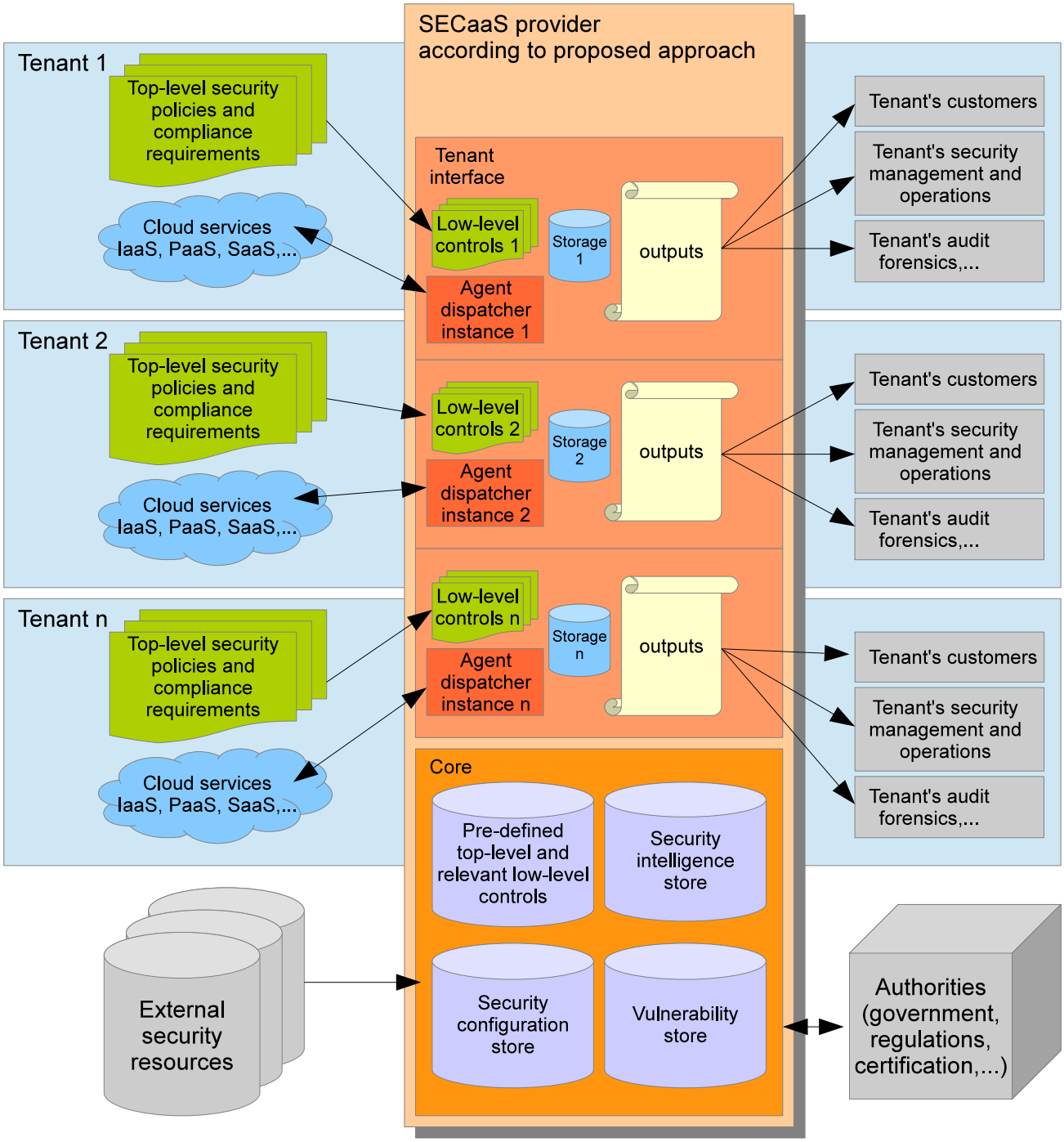

Figure 1: SECaaS provider schema

ISBN: 978-1-61804-134-0