Chapter 8 in Security, Privacy and Trust in Cloud Systems © Springer 2013

Chapter 8

Adaptive Security Management in SaaS Applications

Mohamed Almorsy, Amani Ibrahim, John Grundy

Centre for Computing and Engineering Software Systems
Swinburne University of Technology
Melbourne, Australia
{ malmorsy | aibrahim | jgrundy}@swin.edu.au

Despite the potential benefits, cost savings and revenues that can be gained from adopting the cloud computing model, a downside is that it increases malicious attackers' interest and ability to find vulnerabilities to exploit in cloud software and/or infrastructure. In addition, the cloud model is still not fully mature and lot of issues that impact the model's creditability and pervasiveness are still open. These include vendor lock-in, multi-tenancy and isolation, data placement and management, service portability, elasticity engines, SLA management, and cloud security are well known open research problems in the cloud computing model [1].

Cloud consumers consider security as a major concern that hampers their adoption of the cloud computing model [2]. This is because: (i) enterprises outsource the security management of their cloud-hosted assets to a third party (cloud provider) that hosts their IT assets. This leads to the well-known loss of control problem [3], where cloud consumers do not have security control on their outsourced assets; (ii) co-existence of different tenants' assets in the same location and using the same instance of a software service. This makes them unaware of the soundness of service security and what security controls are used to secure each tenant's data and ensure no data confidentiality breaches exist; (iii) the lack of security guarantees in the Service Level Agreements (SLAs) between cloud consumers and cloud providers. Most existing SLAs focus on reliability, availability and performance of services rather than security [4]; and (iv) hosting valuable enterprise assets on publicly accessible infrastructure increases the possibility of attacks and interest of attackers to exploit vulnerabilities in such services to achieve benefits (such as compromising one tenant's data by other tenants who may be competitors).

From the cloud providers' perspective, security requires a lot of expenditure (e.g. security solutions' licenses), resources (security is a resource intensive aspect), and is a difficult problem to master due to its complexity (in terms of number of services, stakeholders, security solutions, etc.). However, ignoring or delaying the improvement of security in the cloud computing roadmap will not meet expected revenues and take-up. Thus, cloud providers have to understand consumers' concerns and seek out new security solutions that resolve such concerns.

Finally, from the security vendors' perspective, developing different security adaptors to integrate with different cloud services is a big headache for them. The development of a security management framework helps mediate between cloud services and security controls. Such a framework should integrate with security solutions through a common security interface (thus vendors will have to develop only one adaptor), and at the same time integrates with the cloud services to be secured using runtime software instrumentation approaches.

Security management systems help in capturing and defining enterprise asset security, enforcing specified security details, monitoring the security status of these assets, and improving security to meet target security objectives that may also change overtime according to business needs. The security challenges of the cloud computing model make it too hard to depend on manual approaches that require deep involvement of stakeholders, either cloud or service providers or service consumers, to deliver the aimed security level. In order to address cloud security challenges, we have identified five main areas we need to consider in order to deliver an adaptive security management framework for the cloud computing model.

- **Cloud computing:** We need to study the cloud computing model characteristics and what are the main factors that contribute to the cloud computing security problem. We need to identify the key requirements that should be addressed when developing such a security management model for the cloud computing model. This issue was discussed in previous chapters of this book;
- **Security management:** We need to study the existing security management efforts and standards. We need to identify what are the key limitations of these efforts when applied to the cloud computing model, and which one(s) to use or extend when addressing the cloud computing model;

- **Security analysis:** We need to determine what are the main security analysis tasks; what are the existing security analysis efforts that exist in web applications security analysis area; and how far these efforts support automation of the security analysis task. Moreover, we need to know how these efforts are extensible to support discovery of existing as well as new vulnerabilities that emerge at runtime;
- **Security engineering:** We need to capture, inject, and update cloud services' security for different tenants at runtime taking into consideration different SaaS multi-tenancy models. We need to study the existing security engineering efforts; identify key limitations of these efforts that arise from applying these techniques to the cloud services; and how they fit with the multi-tenancy requirements. Moreover, we need to know how much automation is possible with these approaches to facilitate the automated integration of these approaches with the target cloud services;
- **Security monitoring:** We need to determine what security monitoring platforms exist and how these platforms fit into the cloud multi-tenancy model – i.e. how can we capture different tenants' security metrics and how can we plug-in security probes that collect measurements required to assess the security status specified by different tenants.

The remaining part of this chapter is organized as follows. In Section 1, we discuss the key security management standards, differences, and limitations to fit with the cloud computing model. In Section 2, we discuss key efforts in security analysis (as a main source of security requirements). In Section 3, we discuss key efforts in security enforcement. In Section 4, we discuss key efforts and limitations in security monitoring area. In Section 5, we introduce our proposed solution for the cloud computing security management problem and the main framework architecture components. In Section 6, we introduce a structure

8.1 Security Management Standards

Information security management systems [5-7] are defined as systems that deliver a holistic "model for establishing, implementing, operating, monitoring, reviewing, maintaining and improving the protection of information assets". We have identified two key security management standards. The first one is the Federal Information Security Management Act introduced by the National Institute of Standards and Technology - NIST-FISMA [5]. The second one is the International Organization for Standardization and the International Electro-technical Commission - ISO/IEC – ISO27000 [6]. Below we summarize these two standards.

8.1.1 NIST-FISMA Standard

NIST-FISMA was originally declared as an e-Government Act in 2002 [5]. The FISMA standard delivers the guidelines to develop agency-wide security management program that help in categorizing information systems, capturing security objectives, enforcing specified security, and monitoring the security status of agency assets. The standard includes a set of guidelines and standards that help implementing the information security management program as follows:

- Standards for categorizing information and information systems by mission impact – i.e. the impact of security breach on a given information system on the assigned security.
- Standards for minimum security requirements for information and information systems. Based on the security categorization of the information systems, a set of minimum security requirements and security controls baseline is selected from the set of the available security control baselines – i.e. if an information system is assigned a low security impact, this means that the low impact security requirements and controls baseline is selected and should be enforced by the security experts.
- Guidance for selecting appropriate security controls for information systems. Different information systems may have different natures that may require using specific rather than common security controls.
- Guidance for assessing security controls in information systems and determining security control effectiveness. This guideline defines how to select security controls to be assessed, method of the assessment, metrics that could be used, and how the assessment could be conducted.
- Guidance for the security authorization of information systems. This guideline specifies who should be responsible for authorizing the security management plan developed including how the identified risks are addressed /mitigated.
- Guidance for monitoring the security controls and the security authorization of information systems. This guideline defines, for each security controls' family (FISMA standard divides the security controls into a set of 17 security controls families), the set of security metrics that should be measured to monitor the security status of a given system, frequency of applications, nature of the metric, formula of the metric, unit of measure, etc.

8.1.2ISO27000 Standard

The ISO27000 standard [6, 8] provides a model to guide the definition and operation of information systems security management. The ISO27000 targets all types of organizations other than federal agencies as intended in the FISMA standard. The ISO27000 standard has a series of security standards that address different areas in the information systems security management framework as follows:

- ISO 27001: This standard gives an overview of the specification of any ISMS that is based on ISO27000 standard. It shows how the ISMS standard is aligned with the Plan-Do-Check-Act (PDCA) management model. It summarizes the key terminologies existing in the security management process and gives a summary of security controls objectives that should be operated.
- ISO 27002: This standard focuses on security controls' implementation guidance to help organizations during the ISMS implementation, reviewing and authorization phases. It covers how these phases could be done to address different security targets including Human Resources, physical security, communication security, access control, etc.
- ISO 27003: This standard gives guidance on implementation of different ISMS phases including planning processes, do processes, check processes, and act processes phases.
- ISO 27004: This standard addresses the ISMS measurements and metrics that could be used, stakeholders and responsibilities, measurement operations, data analytics of the measurement results, and further improvement actions that could be taken.
- ISO 27005: This standard addresses the security risk management process. It details a methodology for information security risk management including risk analysis, treatment, and acceptance.
- ISO 27006: This standard provides guidelines to help organizations in the accreditation process of ISMS certification. It documents the key requirements that should be satisfied and how they can be addressed.

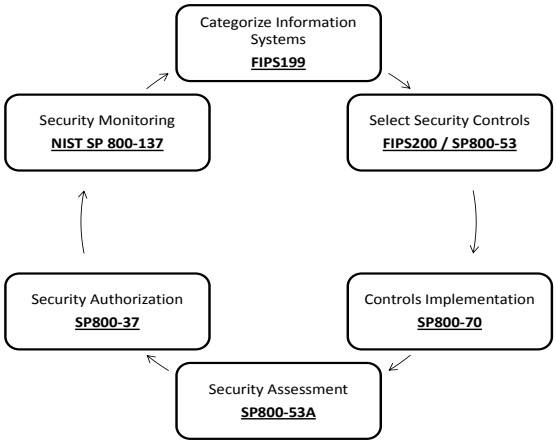

Figure 1. NIST-FISMA main phases, flow, and standards

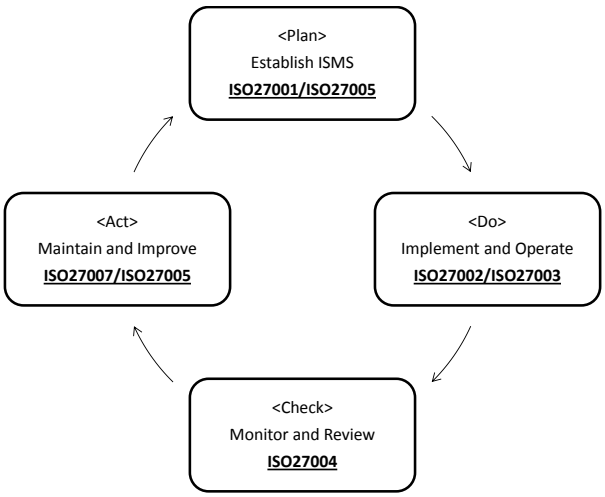

Figure 2. ISO27000 main phases, flow, and standards

8.1.3Differences between NIST-FISMA and ISO27000

We have determined that a lot of similarities exist between ISO27000 and NIST-FISMA standards. This includes the general approach, phases of security management, complexity of both standards to implement and satisfy, relatively similar concepts, and risk management, provide a list of security controls (NIST specifies links to ISO27000 security controls). However, we found a set of key differences between them as well. NIST-FISMA targets federal agencies while ISO27000 target commercial organizations; however, there is no problem to apply NIST standard to commercial organizations. NIST-FISMA focuses mainly on one or more IT systems. On the other hand, The ISO27000 has organizational-wide focus. NIST uses IT systems' categorization as a selector to decide the set of security controls (baseline) to apply, while ISO27000 assumes that the set of security controls provided in the standard are available to be picked up and used according to the situation. In our opinion, this helps in the security controls selection phase by minimizing the scope of security controls to select from (minimize the possibility of error or missing security). Moreover, this security controls baseline could be customized later according to identified and assessed security risks.

8.1.4Security Management Standards and the Cloud Computing Model

Both ISO27000 and NIST-FISMA standards assume that the assets (i.e. IT systems) owner has full control over the security management process of their assets – i.e. these assets are mostly hosted internally inside their network perimeter or at least they can specify and monitor the security of their assets if hosted on a service provider. Thus both standards, in terms of their current specifications, do not fit well with the cloud computing model and the multi-tenancy model where tenants do not have any control on their outsourced assets where service consumers do not have any participation in securing the cloud services . This is a well-known security problem with the cloud computing model known as the "loss-of-control" problem. Multi-tenancy adds a new, complex dimension to the loss-of-control security problem. These security management standards are not designed with taking into consideration the service sharing concept introduced by multi-tenancy – i.e. how to capture, enforce, and monitor service security status for different tenants given that these security requirements may change overtime. Moreover, the set of service tenants evolves at runtime. New tenants may register to use the service at runtime. At the same time, other tenants may unregister from the service.

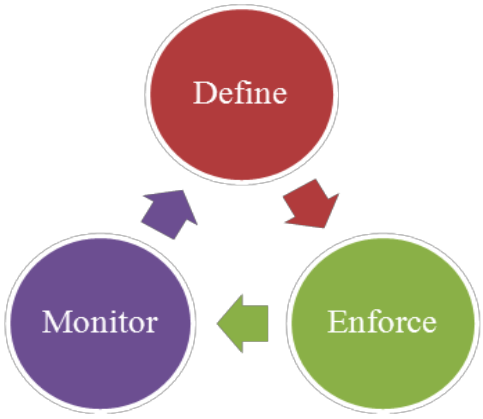

Figure 3. Information security management main phases

8.1.5Rethinking about Security Management Standards

From this analysis of current security management standards, we need to rethink security management systems by considering three key phases: defining security, enforcing security, and monitoring and improving security, as shown in **Figure 3**.

- **Defining Security:** This task focuses on how to identify IT assets to be secured and how to categorize them according to their importance, how to capture different stakeholders' security objectives and goals, how to identify security threats, vulnerabilities and attacks, and how to identify the security requirements and controls that mitigate these (identified) security risks and satisfy these (specified) security objectives.
- **Enforcing Security:** This task focuses on how to implement and configure the specified security controls and how to integrate these security controls within the target IT systems that need to be secured, how to manage changes required to effect new security updates resulting from new business needs and new security risks.

- **Monitoring and Improving Security:** This task focuses on how to capture stakeholders' security metrics, how to generate necessary security probes that can collect measurements of security controls and IT system status, how to analyse these results, and how to improve the operated security to improve the security status to match the specified stakeholders' security objectives.

In the next sections, we discuss each of these building blocks with detailed analysis of the existing efforts and their key limitations, taking into consideration key cloud security challenges, especially service outsourcing and multi-tenancy dimensions.

8.2 Security Analysis

Possible system threats and vulnerabilities represent a key source of security requirements. Security analysis approaches usually depend on (i) *static techniques* that are applied directly to the system source code. This includes pattern matching: searching a "pattern" string inside the source code and the instances where patterns occurred; (ii) *Dynamic techniques* introduce faults in order to test the behaviour of the system under test. Some knowledge about the system is required; and (iii) *Hybrid techniques* that use both static and dynamic analysis techniques to achieve high accuracy and detect complex vulnerabilities.

Attack Analysis is a security analysis area that targets identifying possible attack paths required to achieve a specific attack goal on a target system entity. Chinchani et al [9] propose an approach to represent a possible attack vector on a given target component. The approach is based on modelling a system as a set of entities (targets). These targets are connected to each other if there is a physical link. Each target has a set of information it processes (keys). If the attacker does not know this information then there will be a cost to get it. Thus the attack vector reflects the set of entities that an attacker has to go through in order to reach his target. This approach was introduced as a replacement of attack graphs (action-centric) that suffer from the state explosion problem. Sheyner et al [10] propose an automated approach to generate an attack graph required to achieve a given attack goal. Their approach is based on creating a network model that reflects all atomic attacks existing in each node. They used a modified version of model checking that can generate all possible combinations of possible paths (not only paths that violate or satisfy a given property). They specify the attack goal as CTL expression. The CTL expression is the negation of the attacker objectives. Hewett et al [11] introduce another approach for formal attack analysis that focuses on hosts rather than on network entities. This is another approach to mitigate the state explosion problem arising from using attack graphs. Ou et al [12] introduce an approach for attack graph generation based on logic-programming. This approach represents attack scenarios as a set of data-logs and associated rules.

Threat Analysis is the second side in the security analysis triangle that aims to identify possible threats on a given system using the set of identified vulnerabilities or attack graphs and the system architecture details. We may conduct threat analysis on high level system architecture before the system exists using a set of common known weaknesses in the underlying technologies used. George et al [13] introduce an automated approach for threat identification based on a predefined set of expert-defined rules. Given a UML diagram of the target system along with the set of vulnerabilities existing in a given program, it applies the rules to identify the possible system threats. The approach has a lot of limitations related to the system model and the threat identification rules. Measuring the attack surface of an application is another approach for threat analysis. Pratyusa et al [14] propose a metric for system security based on system attack surface. This approach is based on calculating entry and exist points and the communication channels used. Abi-Antoun et al [15] propose an approach that looks for security flaws in a given system based on the Microsoft STRIDE model. They introduce a set of rules that can be used to identify and decide the possibility of a given threat based on checking rules related to the preconditions of a discovered vulnerability and the existence of sufficient security controls.

Vulnerability Analysis: Existing vulnerability analysis efforts are mostly designed to analyse against specific vulnerability types, such as SQL Injection and Cross-Site Scripting. Jimenez *et al.* [16] review various software vulnerability prevention and detection techniques. Broadly, static program analysis techniques work on the source code level. This includes pattern matching that searches for a given string inside source code, tokens extracted from source code, or system byte code e.g. calls to specific functions. NIST [17] has been conducting a security analysis tools assessment project (SAMATE). A part of this project is to specify a set of weaknesses that any source code security analysis approach should support including SQL

Injection, XSS, OS command Injection, etc. They have also developed a set of test cases that help in assessing the capabilities of a security analysis tool in discovering such vulnerabilities. Halfond *et al.* [18] introduce a new SQL Injection vulnerability identification technique based on positive tainting. They identify "trusted" strings in an application and only these trusted strings can be used to create certain parts of an SQL query, such as keywords or operators. Dasgupta *et al.* [19] introduce a framework for analysing database application binaries to automatically identify security, correctness and performance problems especially SQLI vulnerabilities. They adopt data and control flow analysis techniques as well as identifying SQL statements, parameters, tables and conditions and finally analyse such details to identify SQLI vulnerabilities. Martin et al [20, 21] introduce a program query language PQL that can be used to capture definition of program queries that are capable to identify security errors or vulnerabilities. PQL query is a pattern to be matched on execution traces. They focus on Java-based applications and define signatures in terms of code snippets. This approach limits their capability to locate vulnerability instances that match semantically but not syntactically. Wassermann *et al.* [22] introduce an approach to finding XSS vulnerabilities based on formalizing security policies based on W3C recommendation. They conduct a string-taint analysis using context free grammars to represent sets of possible string values. They then enforce a security policy that the generated web pages include no untrusted scripts. Jovanovic *et al.* [23] introduce a static analysis tool for detecting web application vulnerabilities. They adopt flow-sensitive, inter-procedural and context-sensitive data flow analysis. They target identifying XSS vulnerabilities only. Ganesh et al [24, 25] introduce a string constraint solver to check if a given string can have a substring with a given set of constraints. They use this to conduct white box and dynamic testing to verify if a given system is vulnerable to SQLI attacks. Bau et al [26] perform an analysis of black-box web vulnerability scanners. They conducted an evaluation of a set of eight leading commercial tools to assess the supported classes of vulnerabilities and their effectiveness against these target vulnerabilities. A key conclusion of their analysis is that all these tools have low detection rates of advanced and second-order XSS and SQLI. The average percentage of discovered vulnerabilities is only 53%. Their analysis shows that these tools achieve 87% in detecting session management vulnerabilities and 45% in detecting cross site scripting vulnerabilities. Kals et al [27] introduce a vulnerability scanner that uses a black-box to scan web sites for the presence of exploitable SQLI and XSS. They do not depend on a vulnerability signature database, but they require attacks to be implemented as classes that satisfy certain interfaces. Balzarotti et al [28] introduce composition of static and dynamic analysis approaches, "Saner", to help validating sanitization functions in web applications. The static analysis is used to identify sensitive sources/sinks methods. Dynamic analysis is used to analyse the identified suspected paths.

8.3 Security Enforcement

8.3.1 Security Engineering

Software security engineering aims to develop secure systems that remain dependable in the face of attacks [29]. Security engineering activities include: identifying security objectives that systems should satisfy; identifying security risks that threaten system operation; elicitation of security requirements that should be enforced on the target system to achieve the expected security level; developing security architectures and designs that deliver the security requirements and integrates with the operational environment; and developing, deploying and configuring the developed or purchased security controls. Approaches typically focus on security engineering during system design. Misuse cases [30] capture use cases that should not be allowed and may harm the system operation. UMLSec [31] extends UML with a profile with set of stereotypes to annotate design elements with security requirements. UMLSec provides a comprehensive UML profile but it was developed mainly for use during the design phase. SecureUML [32] provides a meta-model to design RBAC policies of a target system. Both approaches are tightly coupled with the software system design models.

8.3.2 Adaptive Application Security

Adaptive application security is another key area in security engineering that focuses on enabling a given system to adapt its security capabilities at runtime. Extensible Security Infrastructure [33] is a framework that enables systems to support adaptive authorization enforcement through updating in memory authorization policy objects with new low level C code policies. It requires developing wrappers for every system resource that catch calls to the resource and check authorization policies. Strata Security API [34] hosts systems on a strata virtual machine. This enables interception of system execution at the instruction level based on user security policies. The framework does not support securing distributed systems and it

focuses on low level policies again specified in C code. Serenity [35] enables provisioning of appropriate security and dependability mechanisms for Ambient Intelligence systems at runtime. Security attributes are specified on system components at design time. At runtime the framework links such Serenity-aware systems to the appropriate security and dependability patterns. Serenity does not support dynamic or runtime adaptation for new unanticipated security requirements. Morin et al [36] propose a security-driven and model-based dynamic adaptation approach enabling applications to reflect the specified context-aware AC policies. Engineers define security policies that take into consideration context information. Whenever the system context changes, the tool updates the system architecture to enforce the suitable security policies.

8.3.3Multi-tenancy security engineering

Multi-tenant security engineering is a new branch of security engineering. Hong Cai et al [37] proposed an approach to transform existing web applications into multi-tenant SaaS applications. They focus on the isolation problem by analyzing applications to identify required isolation points that should be handled by the application developers. Chang Jie Guo et al [38] developed a multi-tenancy enabling framework based on a set of common services that provides security isolation and performance isolation. Their security isolation pattern considers the case of having different security requirements (for authentication and access control only). However, it depends on the tenant's administration to manually configure security policies, map tenant's users and roles to the application's predefined roles. Pervez et al [39] developed a SaaS architecture that supports multi-tenancy, security and load dissemination. Their architecture is based on a set of services that provide routing, logging, security. Their proposed security service delivers predefined authentication and authorization mechanisms. No control by service consumers of the security mechanisms is supported and no isolation is provided between the authentication and authorization data of different tenants. Menzel et al [40] proposed a model driven approach and language to specify security requirements on web services and web applications composed of web services. Each application instance (and its services) is deployed on a VM. They assume that web applications are composed of web services only, and that multi-tenant security is maintained through using VMs for each tenant (the simplest case of supporting multi-tenancy).

8.4 Security Monitoring and Improving

NIST [41] characterizes security metrics into three types: (i) *Implementation metrics*. These metrics are intended to demonstrate progress in implementing information security solutions and related policies and procedures; (ii) *Effectiveness/efficiency metrics*. These metrics are intended to monitor if the implemented security controls are implemented correctly, operating as intended and meet the desired outcomes. Effectiveness focuses on the robustness of the security controls while efficiency focuses on the capability of the security controls to mitigate the security objectives; (iii) *Impact measures* are used to articulate the impact of IT security on mission including cost savings, public trust. Existing efforts in information security measurements and monitoring focus on proposing guidelines or processes to be followed when defining metrics and collecting measurements. Chandra et al [42] introduce steps to identify the required security metrics in a given system. This includes specify metric requirements, identify vulnerabilities, identify software characteristics, analyse security model, categorize security metrics, specify security metric measures, design metric development process, develop security metric, finalize metric suite. Similar efforts have been introduced for the cloud [43].

8.5 A Collaboration-based Cloud Security Management Framework

8.5.1 Aligning NIST-FISMA with the Cloud Computing Model

To build our adaptive model-based cloud computing security management approach, we found it crucial to base such an approach on well-known and well-defined security management standards, such as ISO27000 or NIST-FISMA. However, such security management standards are far from covering the full complexity of the cloud computing model and mainly multi-tenancy and outsourcing of IT assets. In this Section, we introduce our proposed alignment of the NIST-FISMA standard to fit with the cloud computing model, enabling cloud providers and consumers to be maintain their security management processes on cloud platforms and services. This framework, as summarized in **Table 1**, is based on improving collaboration between cloud providers, service providers and service consumers in managing the security of the cloud platform and the hosted services. Below we explain how we aligned each phase of the NIST-FISMA standard with the cloud computing model.

1. Service Security Categorization

Each service (S_J) hosted on the cloud platform can be used by different tenants. Each service tenant (Ti), or cloud consumer (CC) owns their information only in the shared service (S_J). The tenant is the only entity that can decide/change the impact of

a loss of confidentiality, integrity and availability on their business objectives. Each tenant may assign different impact levels (Low, Medium, or High) to security breaches of their information. NIST has introduced a new project that proposes new model for security management of the cloud computing model - FedRAMP [44]. In FedRAMP, the cloud provider specifies the security categorization of services delivered on their cloud platform. However, this is not sufficient as the cloud provider does not have sufficient knowledge about the impact of information security breaches on their tenants' business objectives. Our approach enables cloud consumers to be involved in specifying the security categorization of their information. Moreover, our approach enables both scenarios where we can consider the security categorization (SC) per tenant or per service. The security categorization of the service is calculated as the maximum of all tenants' categorizations:

$$SC(T_i) = \{(\text{confidentiality, impact}), (\text{integrity, impact}), (\text{availability, impact})\},\ \text{Impact} \in \{\text{Low, Medium, High}\} \quad \text{Eq. (1)}$$

$$SC(S_j) = \{(\text{Confidentiality, Max}\ (\forall\ T_i\ (\text{impact})), (\text{Integrity, Max}\ (\forall\ T_i\ (\text{impact})), (\text{Availability, Max}\ (\forall\ T_i\ (\text{impact}))\ \} \quad \text{Eq. (2)}$$

2. Security Control Selection

The selection of the security controls to be implemented in protecting tenants' assets has two steps: (a) *baseline security controls selection* - the FISMA standard provides a catalogue of security control templates categorized into three baselines (low, medium and high). Based on the security categorization of the tenant or the service we select the initial baseline of controls that are expected to provide the required level of security specified by tenants. (b) *tailoring of the security controls baseline* - we tailor the security controls baseline identified to cover the service possible vulnerabilities, threats, risks and the other environmental factors as follows:

I. The service risk assessment process

- *Vulnerabilities Identification* - this step requires being aware of the service and the operational environment architecture. We consider the involvement of the service provider (SP) who knows the internal structure of the provided service and the cloud provider (CP) who knows the cloud platform architecture.
- *Threat Identification* - the possible threats, threat sources and capabilities on a given service can be identified by collaboration among the SPs, CPs, and CCs. CCs are involved as they have the knowledge about their assets' value and know who may be a source of security breaches.
- *Risk Likelihood* - based on the capabilities of the threat sources and the nature of the existing vulnerabilities, the risk likelihood is rated as low, medium or high.
- *Risk Level (Risk Exposure)* - based on the risk impact (as defined in phase 1) and risk likelihood we drive the risk level as (Risk Level = Impact X Likelihood).

II. The security controls baseline tailoring process

Based on the risk assessment process, the selected security controls baseline can be tailored to mitigate new risks and fit with the new environment conditions (*scoping of the security controls*) as follows:

- Identify the common security controls; the cloud stakeholders decide on which security controls in the baseline they plan to replace with a common security control (either provided by the CPs or by the CCs);
- Identify critical and non-critical system components; the SPs and CCs should define which components are critical to enforce security on it and which are non-critical (may be because they are already in a trusted zone) so no possible security breaches;
- Identify technology and environment related security controls; used whenever required, such as wireless network security controls;
- *Compensating Security Controls* - whenever the stakeholders find that one or more of the security controls in the tailored baseline do not fit with their environment conditions or are not available, they may decide to replace such controls with a compensating control;
- *Set Security controls parameters* - the last step in the baseline tailoring process is the security controls' parameters configuration, such as minimum password length, maximum number of unsuccessful logins, etc. This is done by collaboration between the CPs and CCs. The outcome of this phase is a security management plan that documents service security categorization, risks, and the tailored security controls baseline.

Table 1. Alignment of NIST-FISMA standard with the cloud computing model

Phase	Task	CP	SP	CC	Input	Output
Security categorization	Categorize security impact (SC)	Informed	Informed	Responsible	Business objectives	Security Impact Level
Security controls selection	Register security controls	Responsible	Responsible	Responsible	Control Datasheet	Security controls registry
	Generate security controls baseline	Responsible (Automated by the framework)			Service SC + Controls registry	Controls baseline + matching status
	Assess service risks	Responsible (planned to be automated)			Service + platform arch. + CVE + CWE	Service Vulns + Threats + Risks
	Tailor security baseline	Responsible (planned to be automated)			Baseline + Risk assessment	Security mgmt plan (SLA)
controls implementation	Implement security controls	Responsible (planned to be automated)			Security mgmt plan	Updated Security plan
Security Assessment	Define security metrics	Responsible	Informed	Responsible	Security objective	Security assessment plan
	Assess security status	Responsible (Automated by the framework)			Security assessment plan	assessment report
Service Authorization	Authorize service	Informed	Informed	Responsible	Security plan + assessment report	Service authorization document
Security Monitoring	Monitor security status	Responsible (Automated by the framework)			Security assessment plan	Security status report

3. Security Controls Implementation

The security plan for each tenant describes the security controls to be implemented by each involved stakeholder based on the security control category (common, service specific). The common security controls implementation is the responsibility of the common control provider who may be the CPs (in case of internal security controls) or the CC (in case of external controls). The service-specific security controls implementation is the responsibility of the SPs. Each stakeholder must document their security controls implementation configuration in the security management plan.

4. Security Controls Assessment

Security controls assessment is required to make sure that the security controls implemented are functioning properly and meet the security objectives specified. This step includes developing a security assessment plan that defines: what are the security controls to be assessed; what are the assessment methods to be used; and what are the security metrics for each security control. The results of the assessment process are documented in a security assessment report. This step may result in going back to the previous steps in case of deficiency in the controls implemented or continuing with the next steps.

5. Service Authorization

This step represents the formal acceptance of the stakeholders on the identified risks involved in the adoption of the service and the agreed on mitigations. The security plan and security assessment plan are the security SLA among the involved parties.

6. Monitoring the Effectiveness of Security Controls

The CPs should provide security monitoring tools to help the CCs in monitoring the security status of their assets. The monitoring tools should have the capability to capture the required security metrics and report the collected measures in a security status report either event-based or periodic-based. The results of the monitoring process may require re-entering the SMP to handle new unanticipated changes.

8.5.2 Security Automation

After aligning the FISMA standard with the cloud model we adopted a set of security standards to help improving the framework automation and its integration with the existing security capabilities, as shown in Figure **4** and examples listed in **Table 2**.

- ***Common Platform Enumeration (CPE)*** [45]: The CPE provides a structured naming schema for IT systems including hardware, operating systems and applications. We use the CPE as the naming convention of the cloud platform components and services. This helps in sharing the same service name with other cloud platforms and with the existing vulnerabilities databases such as NVD [46].
- ***Common Weakness Enumeration (CWE)*** *and* ***Common Attack Pattern Enumeration and Classification (CAPEC)*** [45]: The CWE Provides a catalogue of the community recognized software weaknesses. The CAPEC provides a catalogue of the common attack patterns. Each attack pattern provides a description of the attack scenario, likelihood, knowledge required and possible mitigations. We use the CWE and CAPEC as a reference for the cloud stakeholders during the vulnerabilities identification phase.
- ***Common Vulnerability and Exposure (CVE)*** [45]: The CVE provides a dictionary of the common vulnerabilities with a reference to the set of the vulnerable products (encoded in the CPE). It also offers vulnerability scoring that reflects the severity of the vulnerability. We use the CVE to retrieve the know vulnerabilities discovered in the service or the platform under investigation.
- ***Common Configuration Enumeration (CCE)*** [45]: The CCE provides a structured and unique naming to systems' configuration statements so that systems can communicate and understand such configurations. We use the CCE in the security controls implementation phase. Instead of configuring security controls manually, the administrators can assign values to security control templates' parameters. Our framework uses these configurations in managing the selected security controls.

Table 2. Formats of the adopted security standards

Standard	Format	Example
CPE	cpe:/part: vendor : product : version : update : edition: language	cpe:/a:SWINSOFT: Galactic:1.0: update1:pro:en-us
CVE	CVE-Year-SerialNumber	CVE-2010-0249
CWE	CWE-SerialNumber	CWE-441
CAPEC	CAPEC-SerialNumber	CAPEC-113
CCE	CCE-softwareID-SerialNumber	CCE-17743-6

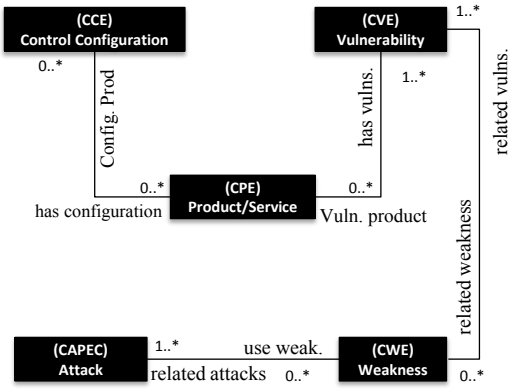

Figure 4. A class diagram of the adopted security standards and their relationships

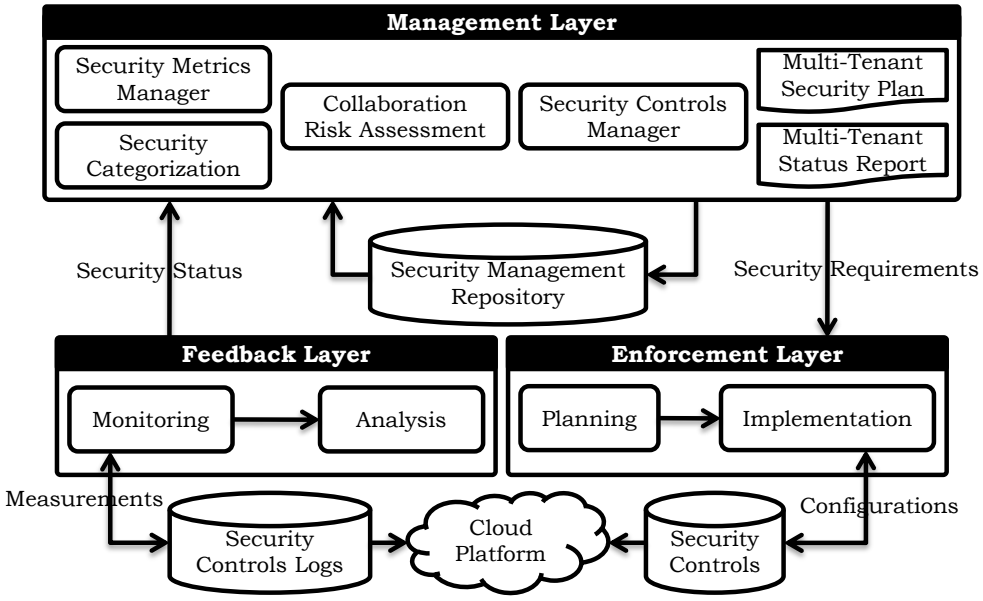

Figure 5. Our collaboration-based security managmenet framework architecture

8.5.3 Cloud Security Management Framework Architecture

Our framework architecture consists of three main layers: a management layer, an enforcement layer, and a feedback layer. These layers, shown in Figure **5**, represent the realization of the ISMS phases.

- ***Management layer.*** This layer is responsible for capturing security specifications of the CPs, SPs, and CCs. It consists of: (a) The security categorization service used by the hosted services' tenants to specify security categorization of their information maintained by the cloud services; (b) The collaborative risk assessment service where all the cloud platform stakeholders participate in the risk assessment process with the knowledge they possess.

(c) The security controls manager service is used to register security controls, their mappings to the FISMA security controls' templates, and their log files structure and locations. (d) The security metrics manager service is used by the cloud stakeholders to register security metrics they need to measure about the platform security. (e) The multi-tenant security plan (SLA) viewer service is used to reflect the tenant security agreement. This shows the tenant-service security categorization, vulnerabilities, threats, risks, the selected mitigation controls and the required metrics. (f) The multi-tenant security status viewer. This reflects the current values of the security metrics and their trends.

- ***Enforcement layer.*** This layer is responsible for security planning and security controls selection based on the identified risks. The selected security controls are documented in the security management plan. The implementation service then uses this plan for maintaining security control configuration parameters and the mapping of such parameters to the corresponding security controls.
- ***Feedback layer.*** This layer has two key services. The monitoring service is responsible for collecting measures defined in the security metrics manager and storing it in the security management repository to be used by the analysis service and by the multi-tenant security status reporting service. The analysis service analyses the collected measures to make sure that the system is operating within the defined boundaries for each metric. If there is a deviation from the predefined limits, the analysis service will give alerts to update the current configurations.

8.6 Usage Example

To demonstrate the capabilities of our cloud computing security framework and our prototype tool implementing this framework we introduce a motivating example, shown in Figure **6**, that happens in any SaaS delivery platform.

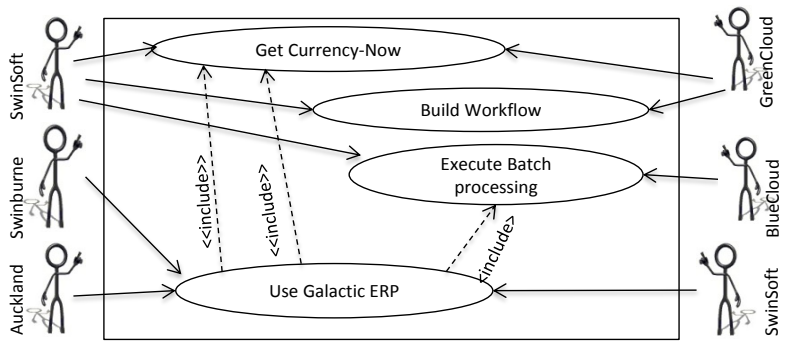

Figure 6. Motivating Example

Consider SwinSoft, a software house developing business applications. Lately, SwinSoft decided to develop a multi-tenant SaaS application "Galactic-ERP". During the development of Galactic, SwinSoft used some of the external services developed and deployed on GreenCloud (a cloud platform that will be hosting Galactic as well) and BlueCloud (a cloud platform hosting some business services). In the meanwhile, SwinSoft has got two tenants (Swinburne and Auckland) who are interested to use Galactic service. Both tenants have their own business as well as their own security requirements. Both of them are worried about the loss-of-control problem arising from the adoption of the cloud model. They would like to maintain their own security requirements on their cloud hosted assets.

The first step in our approach is to register the Galactic ERP service in the cloud platform service repository so that it can be used by the CCs. This step can be done either by SWINSOFT or by GC. In this step we use the CPE name as the service ID, **Figure 7** (top). A new tenant, Auckland, can register their interest in using the Galactic service. Then Auckland will be granted a permission to manage the security of his information maintained by Galactic service. The same is done by Swinburne, **Figure 7** (bottom). Now Auckland and Swinburne can use our framework to maintain their SMP on their assets as follows:

1) ***Service Security Categorization*****:** The Swinburne security administrator specifies the impact level of losing the confidentiality, integrity, and availability of their data maintained by the Galactic ERP service. The same will be done by the Auckland security administrator, as shown in **Figure 7** (bottom). Whenever a new tenant registers their interest in a service and defines their security categorization of data processed by the service (or any of the existing tenants update his security categorization), the framework will update the overall service security categorization.

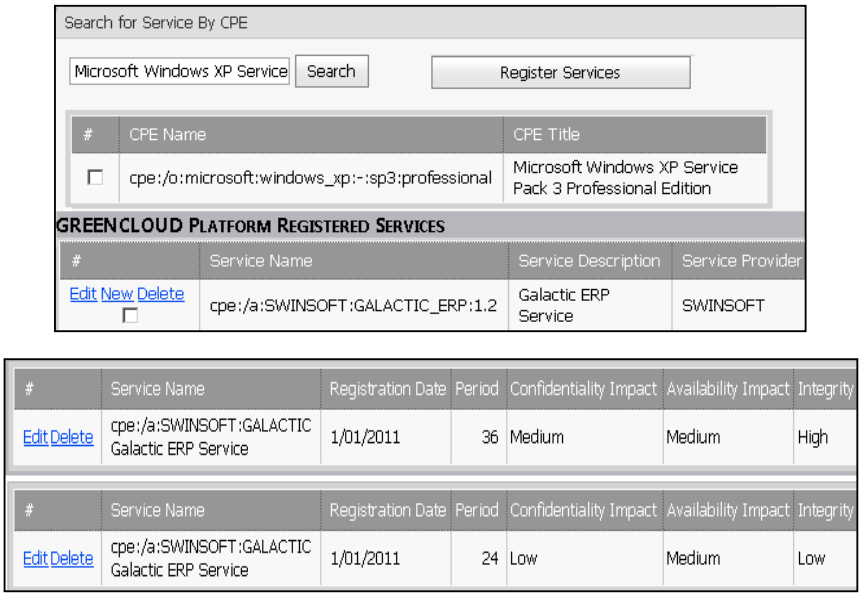

Figure 7. Registering a service (top) and tenants (bottom)

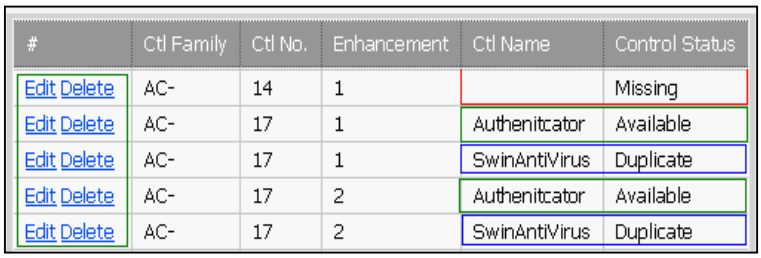

Figure 8. Security controls baseline with controls' status

2) ***Security Controls Selection*****:** GC as a cloud provider already publishes their security controls database. Swinburne and Auckland can register their own security controls using the security controls manager service. Based on the security categorization step, the framework generates the security controls' templates baseline. This baseline identifies the security controls' templates that are: **satisfied** (matches one of the registered security controls), **missing** (does not match registered security controls), and **duplicate** (more than one matched control), shown in **Figure 8.**

a. The Service Risk Assessment Process. Galactic vulnerabilities are identified for the first time by SWINSOFT with the help of GC who know the architecture of the service and the hosting cloud platform. Both SWINSOFT and GC have the responsibility to maintain the service vulnerabilities list up to date. The framework enables to synchronize the service vulnerabilities with the community vulnerabilities database - NVD. Each CC – Swinburne and Auckland – should review the defined threats and risks on Galactic and append any missing threats. The framework integrates with the CWE and CAPEC databases to help stakeholders in identifying possible vulnerabilities whenever the service does not have vulnerabilities recorded in the NVD.

b. The controls baseline tailoring process. The CCs decide which security controls in the baseline they plan to replace with common security controls provided by the CP or the CC, as shown in **Figure 8**. Then SWINSOFT, Auckland, and Swinburne select the critical service components that must be secured. Swinburne and Auckland define their security controls' parameter configurations. The security controls provided by the cloud platform can only be reviewed.

The final outcome of this step is a security management plan that documents the service security categorization, vulnerabilities, threats, risks, and the tailored security controls to mitigate the identified possible security breaches, as shown in **Figure 9**.

3) ***Security Controls Implementation:*** Each stakeholder implements the security controls under their responsibility as stated in the security plan and the security controls configurations as specified before.

4) ***Assessing the implemented security controls*****:** The controls to be assessed and the objectives of the assessment are defined by GC, Auckland and Swinburne, and are documented in the tenant security assessment plan. The execution of such a plan, the assessment process, should be conducted by a third party. Our framework helps in assessing security controls status when using security controls that integrate with our framework (the framework can understand and read their log structure). The outcome of the assessment phase is a security assessment report.

5) ***Service Authorization:*** Swinburne and Auckland give their formal acceptance of the security plan, assessment plan, and the assessment reports. This acceptance represents the authorization decision to use Galactic by the CC.

6) ***Monitoring the effectiveness of the security controls*****:** The framework collects the defined security metrics as per the assessment plan of each tenant and generates status reports to the intended cloud stakeholders. A report shows the metrics status and trends, as shown in **Figure 10**.

The procedure we went through in the example above should be applied not only for published services but also on the cloud platform services themselves. In this case the CP uses our framework to manage the platform security from a consumer perspective. We have done this for the Galactic exemplar used above.

The Security Management plan for the service: Galactic ERP Service

#	Registration Date	Registration (Mths)	Security Categorization
	1/01/2011	24	Low

Vulnerability Name	Vulnerability Description
CVE-2005-0413	Multiple SQL injection vulnerabilities in MyPHP Forum 1.0 allow remote attackers to execute arbitra
CVE-2005-2471	pstopnm in netpbm does not properly use the "-dSAFER" option when calling Ghostscript to convert a
CVE-2005-4195	Multiple SQL injection vulnerabilities in Scout Portal Toolkit (SPT) 1.3.1 and earlier allow remote

Threat Name	Threat Description	Threat Source
DenialSrv	Denial of service	Attacker
InfoCopy	Copy of information at storage	Internal
InfoMod	Modification of information while being transfered	Attacker
MemMod	Modification of data being processed	Maleware

Risk Name	Risk Probability	Confidentiality Impact	Availability Impact	Integrity Impact	Risk Level
DOS	0.7	Low	High	Low	Medium

Control Name	Control Description	Control Baseline	Control Type	Control Family
Authenitcator	an authenitcation security control	Low	Specific	Access Control
SwinAntiVirus	an antivirus security solution	Low	Common	System and Information Integrity
SwinIPS	an intrusion prevention system	Low	CommonControl	System and Information Integrity

Measurement Name	Measurement Description	Frequency	Measrement Steps	Security Control
LoginActivity	Identify the user login rates	48	count(logstatus)	Authenitcator

Figure 9. Auckland security management plan

Measurement Name	Measurement Date	Measurement Value
LoginActivity	12/01/2011	8
LoginActivity	13/01/2011	10
LoginActivity	14/01/2011	14
UnsuccessfulLogins	10/01/2011	2
UnsuccessfulLogins	11/01/2011	4
UnsuccessfulLogins	12/01/2011	6
UnsuccessfulLogins	13/01/2011	8

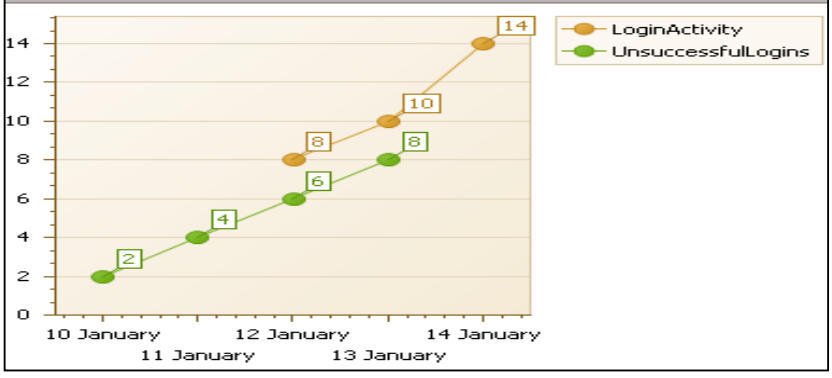

Figure 10. Sample of Swinburne security status report

8.7 Discussion

Our approach provides a security management process; a set of standards-based models for describing platforms, platform services, and services; the security needs of different stakeholders; known threats, risks and mitigations for a cloud deployment; and a tool supporting security plan development and partial automation of a derived security plan. Our approach is comprehensive, supporting all stakeholder perspectives, and collaborative, allowing different stakeholders to develop a mutually-satisfying security model. It addresses the multi-tenancy nature of shared cloud-hosted services when tenants have different security requirements and different SMPs. This is achieved by maintaining and managing multiple security profiles with multiple security controls on the same service. Such controls are delivered by different security vendors. This enables managing traceability between controls and identified risks, and identifies which risks are still not fully mitigated.

The security management process (SMP) of a cloud service has two possible scenarios: either to let each tenant go through the whole SMP as if he is the only user of the service (tenant-based SMP), or to accumulate all tenants' security requirements on a given service and maintain the SMP at the service level (service-based SMP). The later scenario is more straight forward because cloud stakeholders collaborate together to secure the cloud platform and their services with one set of security requirements. The former scenario gives the CCs more control in securing their cloud hosted asset but it has the following problems: (i) the current multi-tenancy feature delivered by cloud services enables tenants to customize service functionality but it does not enable tenants to customize service security capabilities; (ii) the underlying cloud platform infrastructure, such as the VM OS, does not often support multi-tenancy. This means that we cannot install multiple anti-viruses or anti-malware systems on the same OS and be able to configure each one to monitor specific memory processes for a certain user. One solution may be to use a VM for each tenant [40]. However, this work around may not be applicable if the service is not designed for individual instances usage or if the cloud platform does not support VM technology.

Whenever the CCs are not interested in following the security standards or require a light-weight version of our approach, they can leave out as many steps as they want including security controls implementation and customization, security assessment and service authorization steps. The mandatory steps are service categorization and controls selection. Another variation of our framework is to enable CPs to deliver predefined security versions for the service such as service X with (low, medium, high) security profile. CCs can select the suitable version based on their security needs.

8.8 Adaptive Cloud Computing Security Management

The new cloud security management approach we have introduced in the previous sections addresses the loss-of-control and lack-of-trust problems by getting cloud stockholders involved in securing their outsource cloud assets. Our framework is based on the NIST-FISMA standard after aligning it to fit with the multi-tenant cloud model. Moreover, it adopts a set of security standards to automate the security management process. However, our framework lacks two key points: (i) automated integration of security solutions with the target services at runtime without a need for service customization or special preparations at service design time; (ii) automated security analysis of cloud services using an online security analysis service that can analyse services against such vulnerabilities as well as new vulnerabilities at runtime.

Figure 11 shows a more refined version of our framework using models as an abstraction approach. Each stakeholder summarizes their information in models according to their roles. Cloud providers model their platform details, service providers model their service details, and cloud consumers model their security model. These models are weaved in secure-system model (integrated model reflecting critical system entities and security details to be applied on these entities). This model is used to generate a security management plan that guides the configuration of security controls, integration of security controls within the target critical entities either in the service or in the cloud model. In our approach we move from top to bottom in refinement process starting from models to real configurations "Enforcement". On the other side, we collect measure from the services and security controls and consolidate such measures into metrics reflecting security status "Feedback".

Figure 12 shows the high-level architecture of our adaptive-security management framework that we have been working on over the last three years. This security framework should be hosted on a cloud platform and used to manage cloud services security. Our approach architecture is inspired from the MAPE-K autonomic computing [47]:

- **Management Component**

This is a model-based security management component that is responsible for capturing services and security details where service provider system engineers model their services architecture, features and behavior and tenants' security engineers model and verify their own security objectives, requirements, architecture, and metrics. Both models are then weaved together in a tenant secure-system model that guides the next steps of security enforcement and monitoring.

- **Enforcement Component**

This component is responsible for integrating specified security details specified by different tenants with the target cloud services. To support flexible security controls integration with the target services, we developed a common security interface that defines a set of functionalities to be realized by the security vendors through a common security controls adaptor. This enables security controls to easily integrate with our enforcement component which integrates with the cloud services.

- **Monitoring Component**

This component is responsible for generating required security probes according to tenements' specified metrics (captured in the management layer). These probes are then deployed in the cloud services to start capturing measures. Moreover, this component is responsible for collecting the measures from these probes (according to metrics specified frequencies) and passing such measures to the analysis component.

- **Analysis Component**

The analysis component is responsible for two main tasks: performing security analysis of the cloud services including vulnerability and threat analysis. The analysis component analyses the deployed services and their architectures to identify flaws and security bugs. Such issues are delegated to the security management component in order to incorporate in the security status reports for tenants as well as dynamically updating the security controls deployed to block the reported security issues. The analysis component also analyse the measurements reported by the monitoring component against a set of predefined metrics stable ranges – e.g. number of incorrect user authentications per day should be less than 3 trials, so the analysis component should analyse the reported measures of incorrect authentications. This may also include taking corrective actions to defend against such probable attack.

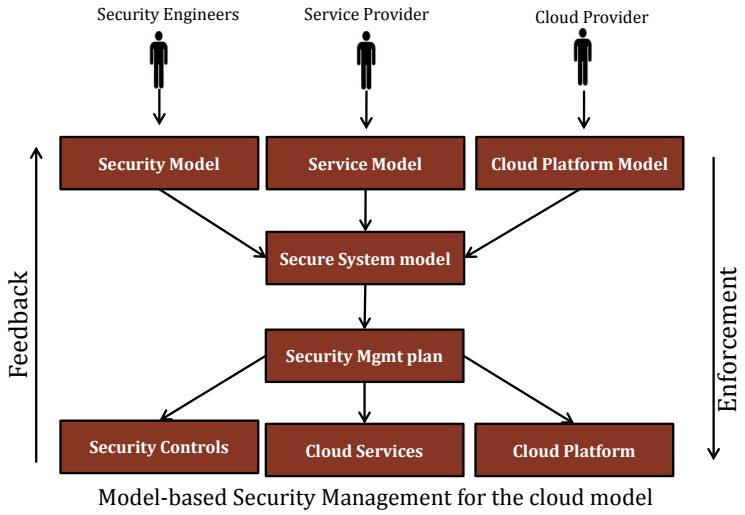

Figure 11. General Approach

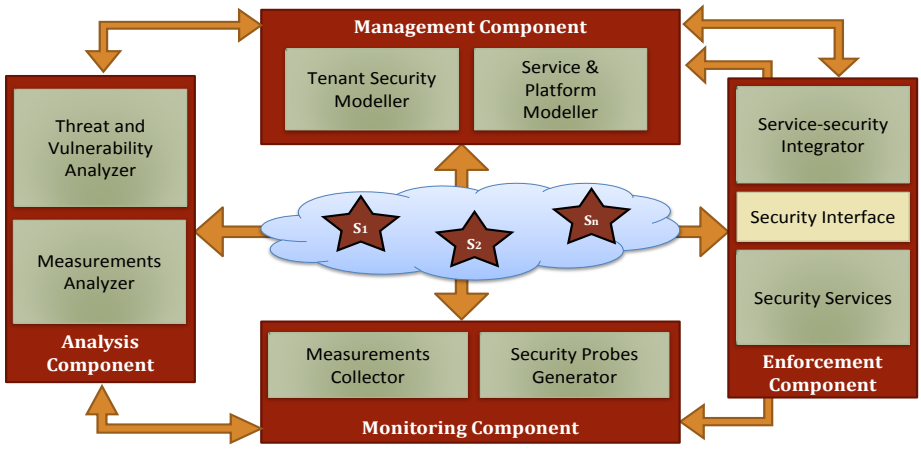

Figure 12. The High-level architecture of our adaptive-security management framework

8.9 Future Research Directions

The area of the cloud computing security is relatively new. Many security problems need to be addressed to promote the trust of the cloud computing model. Here, we summarize three of the key research problems:

- **Data Confidentiality:** Service consumers are worried about their assets (data) security. Thus, they tend to keep their data encrypted while being in transmission, storage, and processing; however, applications need to work on plaintext data. Thus, these applications will need to decrypt customers' data on the cloud platform. This task is prone to attacks from malicious insiders (cloud platform administrators) who have access to the physical servers and may deploy any malicious software to access plain data while being processed in memory. It is highly required to find some approaches that conform that tenants' data confidentiality cannot be breached by malicious insiders.
- **Tenant-oriented Security Engineering:** Multi-tenant applications are shared among different service tenants who may be competitors or malicious tenants. Thus, each tenant is interested to define their own security requirements and enforce their own security controls. Moreover, the set of service tenants emerges at runtime; new customers register to use the service and existing customers may unregister from using the service. Thus, there is a high need to security engineering approaches that help in developing cloud services that can capture, enforce, and integrate different tenants' security requirements and controls at runtime. This also requires developing some security standards that both service providers and security vendors have to follow in order to facilitate the integration between services and required security solutions.
- **Security SLA:** The area of service level agreement becomes one of the hot topics with the wide-adoption of the service outsourcing either as SOA or as cloud computing. However, most of the efforts in the SLA management focus on how to negotiate and define SLA terms including availability, reliability, and performance but not security. Moreover, they focus on how to monitor and avoid violation of the SLA terms. Thus, there is a big need to security SLA management approaches that can define security terms to agree on, monitor the realization and satisfaction of these terms and take proactive and corrective actions whenever needed.

8.10 Conclusion

In this chapter, we introduced a new cloud computing security management model based on joint-collaboration between different cloud platform stakeholders according to who owns the piece of information required to go through the full security management process. This in turn reflects our proposed alignment of the NIST-FISMA standard as one of the main security management standards. We also introduced a usage example of the proposed approach where we have different tenants sharing the same service instance while each stakeholder would like to enforce his security requirements on his cloud hosted assets. We discussed our comprehensive, adaptive security management platform that helps in capturing tenants' security requirements, realizing these requirements and integrating security controls with target cloud services at runtime, and monitoring the security status of these cloud services according to tenants security objectives captured in terms of security metrics.

References

[1] European Network and Information Security Agency (ENISA), "Cloud computing: benefits, risks and recommendations for information security," 2009, http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment, Accessed On July 2010.

[2] International Data Corporation, "IDC Ranking of issues of Cloud Computing model," ed, 2009, http://blogs.idc.com/ie/?p=210, Accessed on July 2010.

[3] Balachandra Reddy Kandukuri, Ramakrishna Paturi and Atanu Rakshit, "Cloud Security Issues," in *Proceedings of the 2009 IEEE International Conference on Services Computing*, 2009, pp. 517-520.

[4] S. A. d. Chaves, C. B. Westphall, and F. R. Lamin, "SLA Perspective in Security Management for Cloud Computing," in *Sixth International Conference on Networking and Services*, Cancun, Mexico, 2010, pp. 212-217.

[5] National Institute of standards and technology (NIST), "The Federal Information Security Management Act (FISMA)," Washington: U.S. Government Printing2002, http://csrc.nist.gov/drivers/documents/FISMA-final.pdf, Accessed on August 2010.

[6] International Organization for Standardization (ISO), "ISO/IEC 27000 - Information technology - Security techniques - Information security management systems - Overview and vocabulary," ISO/IEC 27001:2005(E), 2009, http://webstore.iec.ch/preview/info_isoiec27000%7Bed1.0%7Den.pdf, Accessed On July 2010.

[7] E. Humphreys, "Information security management standards: Compliance, governance and risk management," *Information Security Technical Report,* vol. 13, pp. 247-255, 2008.

[8] A. Tsohou, S. Kokolakis, C. Lambrinoudakis, and S. Gritzalis, "Information Systems Security Management: A Review and a Classification of the ISO Standards," in *Next Generation Society. Technological and Legal Issues*. vol. 26, A. Sideridis and C. Patrikakis, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 220-235.

[9] R. Chinchani, A. Iyer, H. Ngo, and S. Upadhyaya, "A target-centric formal model for insider threat and more," Technical Report 2004-16, University of Buffalo, US2004.

[10] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, 2002, pp. 273-284.

[11] R. Hewett and P. Kijsanayothin, "Host-Centric Model Checking for Network Vulnerability Analysis," in *Computer Security Applications Conference, 2008. ACSAC 2008. Annual*, 2008, pp. 225-234.

[12] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer," presented at the In 14th USENIX Security Symposium, Baltimore, MD, USA, August 2005., 2005.

[13] G. Yee, X. Xie, and S. Majumdar, "Automated threat identification for UML," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, 2010, pp. 1-7.

[14] P. K. Manadhata and J. M. Wing, "An Attack Surface Metric," *IEEE Transactions on Software Engineering,* vol. 37, pp. 371-386, 2011.

[15] M. Abi-Antoun and J. M. Barnes, "Analyzing security architectures," presented at the Proceedings of the IEEE/ACM international conference on Automated software engineering, Antwerp, Belgium, 2010.

[16] A. M. Willy Jimenez, Ana Cavalli "Software Vulnarabilities, Prevention and Detection Methods: A Reviw," in *Proc. of 2009 European Workshop on Security in Model Driven Architecture*, Enschede, The Netherlands, 2009, p. 6—13.

[17] NIST, "Source Code Security Analysis Tool Functional Specification Version 1.1," May 2007, Accessed 2011.

[18] W. G. J. Halfond, A. Orso, and P. Manolios, "Using positive tainting and syntax-aware evaluation to counter SQL injection attacks," in *Proc. of 14th ACM SIGSOFT international symposium on Foundations of software engineering*, Oregon, USA, 2006, pp. 175-185.

[19] A. Dasgupta, V. Narasayya, and M. Syamala, "A Static Analysis Framework for Database Applications," in *Proc. of 2009 IEEE International Conference on Data Engineering*, 2009, pp. 1403-1414.

[20] M. Martin, B. Livshits, and M. S. Lam, "Finding application errors and security flaws using PQL: a program query language," in *Proceedings of the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* CA, USA, 2005, pp. 365-383.

[21] M. S. Lam, M. Martin, B. Livshits, and J. Whaley, "Securing web applications with static and dynamic information flow tracking," in *Proc. of 2008 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation*, California, USA, 2008, pp. 3-12.

[22] G. Wassermann and Z. Su, "Static detection of cross-site scripting vulnerabilities," in *Proc. of 30th international conference on Software engineering*, Leipzig, Germany, 2008, pp. 171-180.

[23] N. Jovanovic, C. Kruegel, and E. Kirda, "Pixy: a static analysis tool for detecting Web application vulnerabilities," in *Proc. of 2006 IEEE Symposium on Security and Privacy*, 2006, pp. 258-263.

[24] V. Ganesh, A. Kieżun, S. Artzi, P. J. Guo, P. Hooimeijer, and M. Ernst, "HAMPI: a string solver for testing, analysis and vulnerability detection," in *Proc. of 23rd international conference on Computer aided verification*, Snowbird, UT, 2011, pp. 1-19.

[25] A. Kieyzun, P. J. Guo, K. Jayaraman, and M. D. Ernst, "Automatic creation of SQL Injection and cross-site scripting attacks," in *Proc. of 31st International Conference on Software Engineering*, 2009, pp. 199-209.

[26] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, "State of the Art: Automated Black-Box Web Application Vulnerability Testing," in *Proc. of 2010 IEEE Symposium on Security and Privacy*, 2010, pp. 332-345.

[27] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovic, "SecuBat: a web vulnerability scanner," presented at the Proc. of 15th international conference on World Wide Web, Edinburgh, Scotland, 2006.

[28] D. Balzarotti, M. Cova, V. Felmetsger, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna, "Saner: Composing Static and Dynamic Analysis to Validate Sanitization in Web Applications," in *Proc. of 2008 IEEE Symposium on Security and Privacy*, 2008, pp. 387-401.

[29] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*: John Wiley and Sons, 2001.

[30] G. Sindre, and A. Opdahl, "Eliciting security requirements with misuse cases," *Requir. Eng.,* vol. 10, pp. 34-44, 2005.

[31] J. Jürjens, "Towards Development of Secure Systems Using UMLsec," in *Fundamental Approaches to Software Engineering*. vol. 2029, ed: Springer Berlin Heidelberg, 2001, pp. 187-200.

[32] T. Lodderstedt, D. Basin and J Doser, "SecureUML: A UML-Based Modeling Language for Model-Driven Security," in *The 5th International Conference on The Unified Modeling Language*, Dresden, Germany, 2002, pp. 426-441.

[33] B. Hashii, S. Malabarba, R. Pandey, and e. al, "Supporting reconfigurable security policies for mobile programs," presented at the Proceedings of the 9th international World Wide Web conference on Computer networks, Amsterdam, The Netherlands, 2000.
[34] K. Scott, N. Kumar, S. Velusamy, and e. al, "Retargetable and reconfigurable software dynamic translation," presented at the Proceedings of the international symposium on Code generation and optimization, San Francisco, California, 2003.
[35] F. Sanchez-Cid, and A. Mana, "SERENITY Pattern-Based Software Development Life-Cycle," in *19th International Workshop on Database and Expert Systems Application*, 2008, pp. 305-309.
[36] Brice Morin, Tejeddine Mouelhi, Franck Fleurey, Yves Le Traon, Olivier Barais, and Jean-Marc Jézéquel, "Security-Driven Model-Based Dynamic Adaptation," presented at the the 25nd IEEE/ACM International Conference on Automated Software Engineering, Antwerp, Belgium, 2010.
[37] H. Cai, N. Wang, and M. J. Zhou, "A Transparent Approach of Enabling SaaS Multi-tenancy in the Cloud," in *Services (SERVICES-1), 2010 6th World Congress on*, 2010, pp. 40-47.
[38] C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, and B. Gao, "A Framework for Native Multi-Tenancy Application Development and Management," in *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on*, 2007, pp. 551-558.
[39] Z. Pervez, S. Lee, and Y.-K. Lee, "Multi-tenant, secure, load disseminated SaaS architecture," in *Proceedings of the 12th international conference on Advanced communication technology*, Gangwon-Do, South Korea, 2010, pp. 214-219.
[40] M. Menzel, R. Warschofsky, I. Thomas, C. Willems, and C. Meinel, "The Service Security Lab: A Model-Driven Platform to Compose and Explore Service Security in the Cloud," in *Services (SERVICES-1), 2010 6th World Congress on*, 2010, pp. 115-122.
[41] M. S. Elizabeth Chew, Kevin Stine, Nadya Bartol, et al, "Performance Measuremenet Guide for Information Security " National Institute of Standards and Technology2008.
[42] S. Chandra and R. A. Khan, "Software security metric identification framework (SSM)," presented at the Proceedings of the International Conference on Advances in Computing, Communication and Control, Mumbai, India, 2009.
[43] J. Bayuk, "Cloud security metrics," in *System of Systems Engineering (SoSE), 2011 6th International Conference on*, 2011, pp. 341-345.
[44] NIST, "Concept of Operations (CONOPS) - FedRAMP," NIST2012.
[45] Mitre Corporation. (2010). *Making Security Measurable*. Available: http://measurablesecurity.mitre.org/
[46] National Institute of Standards and Technology - NIST. (Dec 2010). *National Vulnerabilities Database Home*. Available: http://nvd.nist.gov/
[47] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adapt. Syst.,* vol. 4, pp. 1-42, 2009.