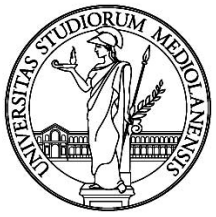

UNIVERSITÀ DEGLI STUDI DI MILANO

DIPARTIMENTO DI INFORMATICA

Corso di Laurea Magistrale in Sicurezza Informatica

Valutazione automatica e continua della compliance in ambienti cloud: Il caso di studio FedRAMP

RELATORE

Prof. Claudio Agostino Ardagna

CORRELATORE

Dott. Marco Anisetti

SECONDO CORRELATORE

Prof. Ernesto Damiani

TESI DI LAUREA DI

Patrizio Tufarolo

Matr. 875041

Anno Accademico 2016/2017

Ai miei genitori e a mio fratello

Acknowledgments

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

.

.

Capitolo 2

Security assurance: lo stato dell'arte e la sfida

2.1 Introduzione

In questo capitolo si approfondirà lo stato dell'arte in materia di **security assurance** e **controllo della compliance**, ovvero la verifica della conformità di un'infrastruttura informatica tradizionale, ibrida o cloud, rispetto a una politica, che può essere sviluppata internamente oppure derivata da un più complesso apparato normativo o da uno standard. In particolare verranno trattate le problematiche di sicurezza introdotte dall'adozione di un approccio *cloud*, all'interno dei processi *IT* di un'organizzazione strutturata sulla base di un'infrastruttura informatica tradizionale.

Spesso si fa coincidere il concetto di cloud computing con quello di outsourcing, di fatto presupponendo che l'adozione di tecnologie *cloud* corrisponda all'attitudine di concedere a terzi gli oneri di gestione di una parte dell'infrastruttura informatica. La definizione di *cloud computing* a cui si fa riferimento in questo elaborato di tesi è quella del NIST[1] nel documento *SP-800-145*[1] nel quale il cloud è presentato come un insieme di tecnologie aventi come obiettivo l'erogazione di servizi e risorse in modalità *on-demand* da un pool condiviso. La condizione di *outsourcing*, quindi, acquisisce una connotazione non strettamente necessaria all'adozione del servizio *cloud*, con cui tuttavia condivide alcuni vantaggi[2] soprattutto per realtà dove il *core business* non è il settore IT:

1. Contenimento dei costi
2. Velocità nel ciclo di sviluppo
3. Garanzia di prestazioni e di qualità
4. Servizio distribuito geograficamente
5. Contratti di affitto strutturati e dimensionati

[1] National Institute of Standards and Technology, *http://www.nist.org/*

2.2 Sicurezza nel cloud computing

Lo scopo finale dell'utilizzo di tecnologie *cloud* consiste nella possibilità per un'organizzazione di usufruire di un modello scalabile, elastico, standard, misurabile e orchestrabile al fine di poter garantire continuità di servizio e prestazioni elevate, demandando la gestione dei processi sistemistici a piattaforme centralizzate e intelligenti. A tal proposito il NIST[1] identifica tre modelli di servizio:

- **IaaS**, *Infrastructure as-a-Service*, nel quale è l'asset erogato è l'infrastruttura informatica, in termini di potenza di calcolo mediante sistemi di virtualizzazione, risorse di rete e storage. Essendo il modello più di difficile gestione, è spesso amministrato tramite un orchestratore. Esempi di tecnologie open-source in questo settore sono *OpenStack*[2], *oVirt*[3], *Apache CloudStack*

- **PaaS**, *Platform as-a-Service*, tramite il quale si fornisce all'utente la possibilità di eseguire servizi personalizzati offrendo meccanismi di contenimento nell'esecuzione, scalabilità e multi-tenancy. L'utente ha un controllo parziale sull'esecuzione del servizio: solitamente egli può interagire in modo limitato con il kernel. Una delle tecnologie più utilizzate è quella dei *container*, un evoluzione del concetto di *jail* proprio dei sistemi operativi BSD, il cui obiettivo è quello di utilizzare meccanismi di segregazione delle risorse basati sulle funzionalità del kernel. I servizi vengono eseguiti in un ambiente isolato, ed hanno un filesystem e uno stack di rete simulato in software dedicati. Una delle implementazioni più note della tecnologia *container* è *Docker*[4] che, nato inizialmente come evoluzione di LXC, è ora basato su una libreria proprietaria e costituisce la base per molte piattaforme PaaS (es. Kubernetes[5] e OpenShift[6]).

- **SaaS**, *Software-as-a-Service*, che permette all'utente di usufruire delle funzionalità di un singolo applicativo, riducendo al minimo l'effort computazionale sulla macchina dell'utente stesso. Tipicamente in questa categoria ricadono le applicazioni web, alcune applicazioni mobile e alcuni software per PC. Alcuni esempi di *SaaS* noti sono *Office 365 Online*[7] e *Google Docs*[8].

[2]Software open-source per la realizzazione di infrastrutture cloud pubbliche e private, https://www.openstack.org/

[3]Software open-source alla base della piattaforma

[4]Docker, *https://www.docker.com*

[5]Kubernetes, soluzione *PaaS* progettata da Google *https://kubernetes.io/*

[6]OpenShift, soluzione *PaaS* di Red Hat *https://www.openshift.com/*

[7]Office 365 è la versione cloud-based della nota suite per l'ufficio *Microsoft Office*, *https://www.office365.com*

[8]Google Docs è una suite per l'ufficio sviluppata da Google ed erogata esclusivamente come applicazione web, *https://docs.google.com*

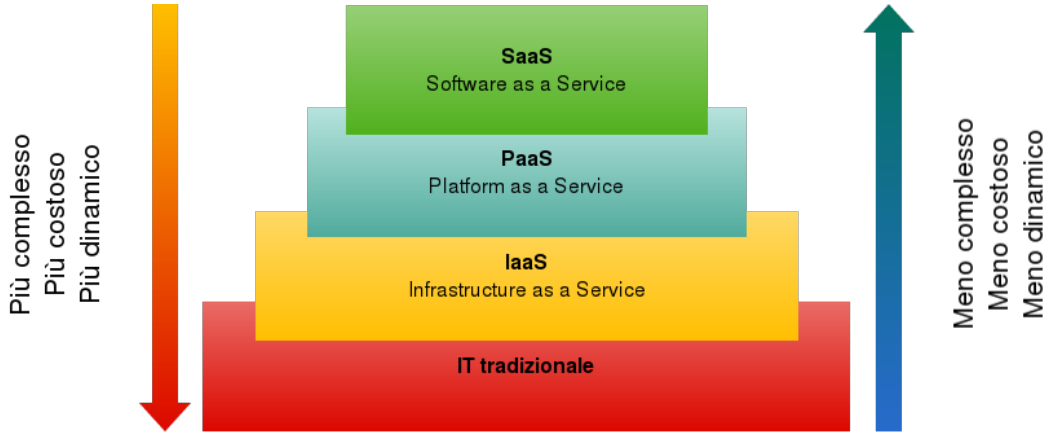

Figura 2.1: Modelli di servizio

La parola chiave è quindi **"automazione"**. Questa, oltre a garantire una solidità del modello di distribuzione di un servizio grazie a schemi dichiarativi, apporta notevoli vantaggi anche dal punto di vista della sicurezza, facilitando la gestione degli aspetti di confidenzialità, integrità e disponibilità.
Il paradigma *as-a-service* ha infatti consentito la costituzione di una *baseline* robusta garantita dalla centralizzazione delle funzionalità di security le quali, essendo erogate come risorse *cloud*, sono interamente gestite dal *cloud service provider* - pubblico o privato - che può demandarne la gestione parziale all'utente mediante meccanismi di orchestrazione, interfacce grafiche ed API.
Se a primo impatto può apparire come un enorme vantaggio, di fatto ciò introduce un *single point of failure*, determinando livelli di rischio aggiuntivi rispetto alle infrastrutture tradizionali. Si pensi, ad esempio, alle funzionalità di *firewalling* offerte generalmente con la denominazione di *security groups* o Firewall as-a-Service (FWaaS): un'implementazione non idonea dal punto di vista funzionale nel substrato infrastrutturale del fornitore di servizi, potrebbe determinare la mancanza di sicurezza per i servizi che ne fanno affidamento. La stessa asserzione è valida per molte altre funzionalità comunemente offerte dal provider: cifratura dei volumi di storage, crittografia e controllo degli accessi nei servizi di block-storage e così via.

Ulteriori riflessioni possono essere fatte anche per quanto riguarda l'aspetto di integrità del dato: se da una parte il cloud service provider implementa già meccanismi di basso livello per la persistenza dello storage, ridondanza, sistemi di backup automatici, dall'altra non si ha la chiara evidenza di come questi aspetti siano effettivamente gestiti e di come la proprietà sia garantita.

Per quanto concerne la proprietà di disponibilità, la dicotomia va ricercata trattando i concetti di disponibilità del dato e disponibilità del servizio separatamente. Il *cloud computing* offre intrinsecamente solidità in quanto basato sui concetti di scalabilità, elasticità e ridondanza. Grazie ai meccanismi di orchestrazione tramite API è infatti possibile configurare le applicazioni per l'*auto-scaling*, al fine di mantenere una qualità adeguata nell'erogazione del servizio al crescere degli utenti. Ciò, dal punto di vista della sicurezza, ha portato a notevoli benefici per quanto riguarda la mitigazione di attacchi DoS[9], garantendo la continuità

[9]Denial of Service

di servizio riducendo i costi. Tuttavia esistono dei prerequisiti per garantire la disponibilità: innanzitutto il *cloud service provider* deve assicurare la ridondanza dei dati e della rete, contemplando l'ipotesi di distribuire le risorse su più località geografiche, con l'obiettivo sia di prevenire guasti localizzati che di erogare la risorsa dalla località più vicina rispetto all'utente.

Nel momento in cui funzionalità comunemente demandate ad hardware specifico vengono implementano in software, si determinano sia benefici che svantaggi che devono sia essere contemplati in fase di valutazione del rischio che trattati nei contratti di service level agreement. Una compromissione dell'interfaccia di gestione della piattaforma cloud, sia che si tratti di una dashboard sia che si tratti di un'interfaccia API, può portare a un'interruzione di servizio.

Gli standard di sicurezza classici, così come l'assetto normativo e i contratti di *service level agreement*, necessitano di essere adeguati per supportare l'integrazione di tecnologie cloud all'interno degli stack tradizionali, tenendo conto delle problematiche di *shared responsability* presentate.

Il NIST [1] riconosce quattro diversi modelli di deployment:

- **Public Cloud**: modello in cui le risorse sono fornite per un utilizzo pubblico. È tipicamente erogato in outsourcing tramite la rete internet. L'hardware è in mano a un unico provider che eroga servizi in *outsourcing* e ne dispone le metriche e la tariffazione.

- **Private Cloud**: cloud dedicata a un'azienda o organizzazione, sfruttata per erogare servizi appartenenti al provider. L'hardware è generalmente nel datacenter dell'organizzazione.

- **Hybrid Cloud**: approccio ibrido dato dalla composizione di public cloud e private cloud, o di public cloud e infrastrutture tradizionali. Le infrastrutture coinvolte rimangono distinte e sono legate tra loro da un'unica tecnologia (standard o proprietaria) che facilita la migrazione e la portabilità delle risorse.

- **Community Cloud**: modello che fornisce una cloud per uso esclusivo di una comunità di utenti appartenenti ad organizzazioni con obiettivi funzionali comuni. Può essere di proprietà di una o più organizzazioni della community, o di terze parti.

Per ognuno di questi modelli è possibile esplicitare dei requisiti da soddisfare al fine di colmare il rapporto di sfiducia proprio di questo settore[3].

2.3 Valutazione del rischio: vulnerabilità, minacce e attacchi

In letteratura sono stati proposti molti lavori sulla valutazione del rischio su infrastrutture cloud. Nei paragrafi a seguire verranno discussi alcuni di questi approcci, sulla base della metodologia utilizzata da Ardagna et Al.[3]. Le vulnerabilità possono essere categorizzate in tre macro aree, in base alla superficie di attacco considerata:

1. **Livello applicativo**: quando l'attacco è condotto da un qualsiasi attore nei confronti di una piattaforma SaaS

2. **Tenant su tenant**: quando l'attacco è condotto da attori appartenenti a un tenant nei confronti di un altro tenant

3. **Provider su tenant** e **Tenant su provider**: quando l'attacco è condotto dal provider nei confronti di un tenant (tipicamente malevolo) oppure da un tenant nei confronti del provider

2.3.1 Livello applicativo

Si tratta di vulnerabilità tradizionali che da anni tengono sotto scacco il panorama *web services*: si va da attacchi protocollari sulla comunicazione tra servizi fino alla compromissione di applicativi software specifici. Il target dell'attacco sono le piattaforme SaaS, spesso derivate dal porting di un'applicativo tradizionale sul cloud e non nativamente pensate per essere erogate online: per questo motivo sono caratterizzate da una superficie di attacco molto vasta.

Alcuni lavori significativi citati nel survey di rifermento [3] sono:

- **Gruschka and Iacono, 2009**[4], nel quale è stato presentato un *replay attack*, sfruttando una vulnerabilità del meccanismo di verifica della firma digitale sull'interfaccia SOAP di *Amazon EC2*, e sono state eseguiti comandi sulle API con i privilegi di un utente legittimo

- **Bugiel et Al., 2011**[5], che hanno analizzato le minacce sulla confidenzialità e la privacy estraendo con successo informazioni sensibili da immagini di macchine virtuali Amazon

2.3.2 Tenant su tenant

Le vulnerabilità *tenant su tenant* sono tipiche dei sistemi virtualizzati, quando tenant differenti condividono la stessa infrastruttura e, più specificatamente, lo stesso hardware fisico: gli attacchi possono avvenire per configurazioni erronee o vulnerabilità sull'infrastruttura di virtualizzazione. Si tratta quindi di attacchi che avvengono al livello più basso dello stack cloud[3].

Alcuni contributi interessanti per questa categoria di attacchi e vulnerabilità sono quelli di:

- **Ristenpart et al**, 2009[6], in cui è discusso un attacco alla confidenzialità delle informazioni relative a istanze di servizi in esecuzione. L'attacco dimostrato è basato sul fatto che i servizi sono ospitati sullo stesso hardware, per cui per un servizio è possibile generare traffico e monitorare le proprie performance per fare inferenza su quelle di un altro servizio.

- **Green**[7], che propone un ulteriore attacco di tipo *side-channel* che coinvolge, questa volta, due virtual machine ospitate sullo stesso hardware.

2.3.3 Provider su tenant, tenant su provider

Le vulnerabilità di questo tipo si verificano ogni qual volta un utente o un'organizzazione sposta le proprie risorse su un'infrastruttura cloud non fidata - nella quale il provider è malevolo oppure semplicemente curioso - oppure nel caso in cui l'utente inizia ad usare un servizio cloud con l'obiettivo di attaccare il provider (ad esempio creando botnet per lanciare attacchi denial of service, attaccando le API di orchestrazione e così via) [3].

Le tipologie di attacchi che sfruttano queste vulnerabilità, sono generalmente rivolte al livello IaaS[3], ma non è esclusa la possibilità di attacchi a livello PaaS e SaaS.

Il survey si sofferma sul lavoro Huan Liu[8], il quale illustra una attacco DDoS basato sulla saturazione della banda della rete virtuale: la virtualizzazione dello stack di rete a livello software (*software-defined network*) richiede, oltre a risorse di rete, anche un'elevata capacità di calcolo.

Le vulnerabilità provider su tenant sono invece trattate da Rocha e Correira[9] che propongono una panoramica dei possibili attacchi alla confidenzialità - che possono essere condotti anche dal fornitore di servizi - discutendone le contromisure, e da Bleikertz et al. [10] che si concentrano sulla problematica di proteggere i clienti da attacchi condotti da provider esterni, fornendo un'architettura *Cryptography as-a-Service client-driven*.

La problematica di confidenzialità nella casistica *provider-on-tenant* è anche l'oggetto di De Capitani di Vimercati et. al[11] in cui è descritta una tecnica per preservare la confidenzialità del dato riallocandolo in modo dinamico ad ogni accesso su tre nodi, risolvendo così anche i problemi di collusione tra i service provider coinvolti. Un ulteriore articolo di De Capitani di Vimercati et al[12]. affronta la problematica del provider *onesto ma curioso* con una soluzione per l'integrità dei risultati delle query di join, che discute la casistica di un server di storage di terze parti e di fornitori di potenza di calcolo esterni e malevoli i quali producono i risultati del join per basi di dati ospitate esternamente.

2.4 Tecniche di sicurezza per la cloud

Data l'etereogeneità delle problematiche e degli approcci adottati negli articoli citati, è possibile affermare che garantire proprietà di sicurezza in ambienti cloud è molto impegnativo: questi lavori presentano solamente soluzioni parziali al problema, affrontando di volta in volta problemi specifici e presentando tecniche sviluppate *ad-hoc*[3]. Saranno di seguito presentati alcuni approcci e tecniche per garantire la sicurezza su sistemi cloud.

2.4.1 Autenticazione e controllo degli accessi

I sistemi tradizionali per l'autenticazione e il controllo degli accessi si sono verificati inefficienti per la cloud, pertanto è stato necessario definire nuovi approcci. L'adozione di nuovi *pattern* di sviluppo orientati alla scalabilità - come ad esempio il pattern *micro-services*, naturale evoluzione delle architetture SOA - ha reso necessario sviluppare meccanismi di autenticazione decentralizzati e federati.

Almulia and Yeun [2010] offrono una panoramica sui protocolli di autenticazione e *identity management*, analizzandone la sicurezza, l'effort implementativo e i costi[13].

Costituendo parte critica per la maggior parte dei sistemi, i servizi di autenticazione, gestione dell'identità e gestione delle policy di accesso sono erogati *as-a-service*.

Takabi e Joshi hanno descritto un sistema di gestione delle policy *as-a-service* (PMaaS, *Policy Management as-a-service*) che fornisce un punto di controllo centralizzato indipendente dalla locazione della risorsa[14]. Prima di accedere a una risorsa è necessario contattare il server di autenticazione e autorizzazione centralizzato che rilascerà il *grant* dopo opportuna verifica. *Azure Active Directory*, il porting *SaaS* di Microsoft Active Directory, provvede sia a funzionalità di autenticazione che di policy management e fornisce alcuni driver di integrazione per la maggior parte dei protocolli noti.

Tuttavia, poiché molte realtà complesse dispongono già di meccanismi di autenticazione mediante *ticket granting* isolate dalla rete Internet, sono stati ideati anche modalità di autenticazione e controllo degli accessi completamente *stateless* (ad esempio OAuth). È il caso dei *JSON Web Token*, formalizzati nella RFC 7519: l'*authentication server*, dopo aver validato la richiesta di autenticazione, restituisce un token JSON firmato che contiene l'identità dell'utente e tutti i *grant* per le autorizzazioni ad esso relative. Non esiste il concetto di sessione, la validità del token è data esclusivamente da una marca temporale e da una durata. Il token può essere utilizzato quindi per autenticare le richieste verso i vari servizi, cui spetta l'onere di verificarne la validità del contenuto e della firma, decifrabile tramite segreto condiviso con il server di autenticazione che lo ha emesso. I vantaggi di un approccio simile sono molteplici, tuttavia è impossibile revocare il token una volta emesso. Eventuali blocchi sono effettuabili tramite sistemi di *blacklisting* che riporterebbero in auge la problematica della decentralizzazione che si voleva risolvere. La prassi è quindi quella di emettere token one-time o con durata breve, al fine di minimizzare la durata di una possibile finestra temporale di attacco.

2.4.2 Crittografia, firma digitale e trusted computing

La crittografia è essenzialmente utilizzata per proteggere la confidenzialità dei dati, delle comunicazioni e le attività sensibili da tutti quegli avversari che mirano a disturbare l'operatività della cloud. La maggior parte della letteratura utilizza tecniche di crittografia per preservare la confidenzialità: l'obiettivo di queste metodologie è di facilitare la migrazione dei dati gestiti da sistemi tradizionali verso la cloud. Tuttavia non sono assenti tecniche focalizzate su altre proprietà di sicurezza, come l'utilizzo della firma digitale per curare gli aspetti di integrità e privacy.

Trusted Computing

Il *trusting computing* è una tecnica utilizzata per effettuare computazioni sicure, basata sull'utilizzo della crittografia asimmetrica e di un dispositivo hardware

dedicato (TPM, Trusted Platform Module) tramite il quale è possibile *i)* identificare univocamente i dispositivi con un numero di serie e una chiave di cifratura implementata in hardware *ii)* cifrare informazioni con la chiave di cifratura *iii)* firmare informazioni con la chiave di cifratura. Queste funzionalità pongono le basi per una serie di utilizzi avanzati volti a preservare l'integrità e la confidenzialità di dati - sia in transito su una rete, che memorizzati su disco o sui firmware del dispositivo - codice e hardware, riducendo o annichilendo gli effetti di eventuali attacchi.

Boampong e Wahsheh nel 2012 hanno proposto un modello per utilizzare il TPM al fine di garantire la correttezza dei processi di autenticazione, l'integrità e la confidenzialità sulla cloud[15]. Portare il TPM sul cloud significa realizzarne una versione virtuale, così come illustrato da Krautheim[16] nel 2009, basandosi sul concetto di virtual-TPM (vTPM) già descritto da Berger et al. nel 2006[17]. Il vTPM è un componente software che implementa le stesse funzionalità del TPM hardware, garantendo la multi-tenancy mediante istanze multiple e multiplexing. I vantaggi dell'utilizzo di una tecnologia di *trusted computing* nel contesto cloud sono molteplici, come la possibilità per l'utente di fare enforcement di politiche di privacy togliendo la possibilità al cloud service provider di modificarle, fornendo una soluzione parziale problematiche di *shared responsability* discusse. Come illustrato da Velten and Stumpf[18] e più recentemente da Szefer e Lee[19] il TPM può essere utilizzato per garantire confidenzialità e integrità a tutti i livelli dello stack, prevenendo tampering da parte del fornitore di servizi e attacchi da parte di altri tenant o da malware.

2.5 Approcci per assurance, testing, monitoraggio e compliance

I progressi nella ricerca sulla sicurezza della cloud hanno portato la necessità di avere tecniche di *security assurance* per aumentare la confidenza degli utenti nei confronti del provider[20]. Per *assurance* si intende la modalità per ottenere, con un certo livello di precisione, la consapevolezza che l'infrastruttura e/o le applicazioni manterranno nel tempo una o più proprietà di sicurezza, e la loro operatività non sarà compromessa indipendentemente da malfunzionamenti o attacchi[21]. In accordo con Ardagna et Al.[3], è possibile affermare che quello di *assurance* è un concetto più esteso della mera nozione di *sicurezza informatica*, comunemente definita come *la protezione delle informazioni e dei sistemi informativi da accessi, utilizzi disclosure, interruzioni del funzionamento, modifiche e distruzioni non autorizzate*. Nella cloud è molto facile avere livelli di sicurezza elevati con livelli di assurance scarsi poiché le funzionalità di sicurezza realmente implementate sono difficilmente percepite.

Per la messa sicurezza delle realtà che decidono di trasferire degli *asset* sulla cloud è necessario considerare tre aspetti fondamentali:

- Necessità di una soluzione di analisi e gestione del rischio, in grado di valutare l'impatto dell'adozione di servizi cloud sul business

- Esigenze di *transparency*, ovvero la possibilità per l'utente di essere consapevole del modello di business del fornitore di servizi
- Soluzione di assessment, verifica delle policy e della compliance, che permetta sia di verificare lo stato istantaneo del livello di conformità, che di spiegare all'utente le metodologie attuate per mantenere livelli di compliance adeguati, secondo il principio "comply-or-exmplain" di MacNeil and Li[22]

L'obiettivo di questo lavoro di tesi è quello di fornire un framework per la security assurance i) insistendo sulla valutazione continuativa dello stato di sicurezza sulla cloud ii) offrendo un framework cloud-based per la security assurance insistendo su

- testing di proprietà non funzionali
- monitoraggio continuativo della sicurezza del sistema
- conformità del sistema a politiche di sicurezza, siano esse definite internamente ad un'organizzazione, siano esse provenienti da uno standard di settore
- ottemperare alle esigenze di transparency degli utenti della cloud, offrendo una dashboard panoramica sullo stato della cloud del provider

2.5.1 Testing di proprietà non funzionali

Il *testing* è definito come la fase del ciclo di vita del software composta da tutte le attività, statiche o dinamiche, atte a determinare che questo soddisfi i requisiti specificati e che sia conforme all'obiettivo proposto, nonché per rilevare eventuali difetti.

Nel contesto *cloud* possiamo riconoscere due tipologie di soluzioni di testing: quelle specifiche per il collaudo di infrastrutture cloud e quelle generiche per il testing del software, applicabili anche a servizi cloud.

Il lavoro di tesi si focalizzerà maggiormente sulla prima categoria insistendo sulla validazione delle proprietà a tutti i livelli dello stack (in accordo con Riungu et. Al[23]); nonostante ciò il framework proposto può essere adattato ad entrambe le tipologie.

2.5.2 Monitoraggio continuativo della sicurezza del sistema

La natura stessa dei sistemi cloud complica notevolmente l'analisi delle informazioni relative allo stato dei servizi: a causa dell'elevata complessità dei software impiegati nell'orchestrazione e nell'erogazione delle risorse è spesso difficile rilevare cambiamenti nello stato del sistema, il cui back-end è continuamente tempestato di eventi. È quindi necessario introdurre una componente di monitoraggio, collezionamento e correlazione di eventi.

Per valutare aspetti non funzionali come la sicurezza, è poi necessario che questi eventi vengano contestualizzati: possono essere necessarie pertanto analitiche *stateful*, effettuabili anche tramite strumenti più complessi o provenienti dal mondo *big-data*.

Proprio per facilitare scenari di *software integration* il framework proposto nei prossimi capitoli è stato strutturato esasperando la modularità, ed è stato basato principalmente su tecnologie *open-source*.

Come per il testing, anche per il monitoraggio è possibile individuare sia soluzioni generiche sia soluzioni specifiche per il mondo *cloud*. Software come *Nagios*[10] e *Ganglia*[11] rientrano nella prima categoria, ma vantano livelli di espandibilità tali da poter essere adeguati ai sistemi di collezionamento delle metriche dei maggiori software cloud. Ulteriori soluzioni come *Sensu*[12], *Sysdig*[13], *Weave*[14] contengono strumenti specifici per la cloud.

Per quanto riguarda gli aspetti di sicurezza, la disponibilità di potenza computazionale on-demand, ha garantito la possibilità di effettuare il deploy scalabile di sistemi IDS[15] e IPS[16]. L'utilizzo di questa tipologia di software è stato approfondito da Modi et al. [24], i quali hanno illustrato come utilizzarli sulla *cloud* al fine di mitigare le diverse tipologie di attacchi al paradigma CIA (attacchi provenienti dall'interno, dall'esterno, attacchi di flooding, *privilege escalation*, *port scanning*, attacchi agli *hypervisor* di virtualizzazione e attacchi tramite *backdoor*). Un lavoro di Ficco et Al. del 2013[25] ha presentato un'architettura multi-layer per il rilevamento delle intrusioni, che supporta l'aggregazione di eventi complessi.

Lavori successivi hanno successivamente presentato approcci più specifici e focalizzati su problemi singoli, come Ardagna et Al. 2014[20] che tramite un approccio introspettivo sulle virtual-machine ha prodotto un meccanismo di rilevazione dei *rootkit*.

2.5.3 Conformità del sistema a politiche di sicurezza e cloud transparency

Il presente lavoro di tesi trova le sue origini nel progetto europeo FP7 CUMULUS[26] (Certification infrastrUcture for MUlti-layer cloUd Services), nel quale sono stati proposti modelli, processi e strumenti a supporto di un processo di certificazione per proprietà di sicurezza e non-funzionali in ambito di cloud computing. L'obiettivo del processo di certificazione è quello di fornire quante più evidenze possibili per attestare che un sistema software garantisca determinate proprietà non funzionali e si comporti in modo corretto[3]; si tratta di un approccio alla sicurezza già sperimentato in altri ambiti che tuttavia rimane di difficile applicazione nel contesto dei *web-services*, in particolare nella cloud[27]. Infatti, le tecniche di certificazione usuali che considerano il software come blocco monolitico, vanno a scontrarsi con una struttura complessa e *multi-tier*[27] e necessitano

[10]Nagios, piattaforma di monitoraggio distribuita general purpose, *http://www.nagios.org/*

[11]Ganglia, soluzione per il monitoraggio delle performance dei cluster in ambito grid computing, *http://ganglia.sourceforge.net*

[12]Sensu, *https://www.sensuapp.org/*

[13]Sysdig, sistema per l'identificazione dei problemi nei sistemi basati su containerhttp://www.sysdig.org

[14]Weave, piattaforma SaaS per il monitoraggio di architetture a micro-servizi *https://www.weave.works/*

[15]*Intrusion Detection System*, software per il rilevamento delle intrusioni

[16]*Intrusion Prevention System*, sistemi preventivi per la rilevazione di attività anomale usati per prevenire incidenti informatici

di essere integrate con i processi e caratteristiche tipiche del mondo cloud, come il deployment, la discovery degli asset, l'elasticità e il paradgima on-demand. Il problema è stato dapprima affrontato in Damiani et al. [2009b] [28], in cui è definita una soluzione di certificazione per i servizi basata su certificati di sicurezza basati su test-case firmati. Più recentemente Anisetti et. al [29][**CertSoa**][30] hanno proposto uno schema di certificazione sulla base di un processo di testing basato su un modello, esteso poi con un processo di certificazione incrementale al fine di coprire le esigenze evolutive del paradigma dei servizi.

L'obiettivo di questa tesi, tuttavia, non è quello di fornire un meccanismo di certificazione, bensì quello di offrire un framework per il controllo della conformità di un sistema rispetto alle proprietà non funzionali attese. La metodologia utilizzata è basata sul concetto di *auditing*, ovvero la possibilità di verificare il comportamento di un sistema per valutarne l'adeguatezza rispetto alle policy dell'utente piuttosto che ai regolamenti o alle legi vigenti.[3] Si vuole quindi rendere la cloud *auditable* - al fine di ottemperare alle esigenze di transparency dell'utente, incrementando così il livello di fiducia dell'utente nei confronti del provider e permettendo allo stesso di essere in grado di effettuare scelte ponderate delle varie soluzioni rispetto ai propri requisiti, funzionali e non.

La *transparency* consiste, per l'appunto, nel concedere all'utente una visione di alto livello a dati aggregati ed evidenze collezionati dal provider stesso a basso livello, ed è considerato alla base di ogni approccio efficace per la cloud assurance[20][31].

L'assenza di *transparency* infatti rende i problemi di sicurezza difficilmente percettibili per l'utente[3], in quanto i contratti di *service level agreement* non forniscono parametri tecnici per misurare il livello di sicurezza delle applicazioni e dei dati ospitati sulla cloud[32].

Essa, inoltre, è fondamentale per supportare sia una visione dei processi interni da parte del provider che una visione dei processi esterni per il cliente per fini di sicurezza, così da bilanciare entrambe le esigenze[3].

2.6 Conclusioni

Finora la *security assurance* è effettuata mediante tecniche perlopiù manuali e dall'effort elevato, con cadenze trimestrali o semestrali: il processo di verifica non è effettuato con continuità.

Si vuole perciò realizzare un framework automatico e programmabile per documentare, valutare, osservare dei controlli tecnici (auditing su controllo degli accessi, configurazione del sistema, crittografia ecc.), controlli di processo (analisi delle vulnerabilità, analisi del rischio, acquisizione di evidenze sul funzionamento di sistemi e servizi) e controlli di sistema (gestione delle configurazioni, consapevolezza e training, gestione delle modifiche e dei cambiamenti)

Nei prossimi capitoli verrà illustrato FedRAMP, un programma governativo americano che fornisce un approccio standard per effettuare il *security assessment* e automatizzare il monitoraggio continuativo dei servizi cloud; verra mostrata la soluzione Moon Cloud per il controllo della conformità di infrastrutture, piattaforme e software sulla *cloud* basato su metodologie di testing e monitoraggio,

saranno illustrate le implementazioni di alcuni *security control* di FedRAMP in Moon Cloud per concludere con la valutazione e la validazione del framework mediante l'assessment del deployment di un'architettura software orientata a microservizi in modalità multi-layer.

Capitolo 3

FedRAMP - Federal Risk and Authorization Management Program

3.1 Introduzione

.

.

3.2 Struttura

.

.

.

.

3.3 FedRAMP readiness

.

.

.

.

3.4 Controlli di sicurezza per la conformità

.

.

.

.

3.5 FedRAMP in Amazon AWS

.

.

Capitolo 4

Moon Cloud: un framework per il monitoraggio e la verifica della sicurezza

4.1 Introduzione

.

4.2 Architettura e componenti

.

.

.

.

4.3 Moon Cloud come strumento di verifica della compliance

.

4.3.1 Controlli di sicurezza

.

.

.

.

4.3.2 Regole di valutazione

.

.

.

.

Capitolo 5

Implementazione dei controlli di sicurezza FedRAMP in Moon Cloud

5.1 Controlli automatici

5.1.1 Una sezione per ogni security control implementato

5.2 Controlli ad interazione umana

5.2.1 Questionari per l'assessment dei processi di business

Capitolo 6

Validazione del framework

6.1 Deployment di Moon Cloud

6.2 Sicurezza del deployment

6.2.1 Caratteristiche del deployment

6.2.2 Confidenzialità

6.2.3 Integrità

6.2.4 Disponibilità e affidabilità

6.2.5 Data remainance

6.3 Scalabilità e Prestazioni

6.3.1 una sezione per ogni security control eseguito

Capitolo 7

Conclusioni e sviluppi futuri

Bibliografia

[1] National Institute of Standards e Technology (Peter Mell Timothy Grance). *The NIST Definition of Cloud Computing*. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[2] S. Dhar. «From outsourcing to Cloud computing: Evolution of IT services». In: Technology Management Conference (ITMC), 2011 IEEE International. IEEE.

[3] Claudio A. Ardagna et al. «From Security to Assurance in the Cloud: A Survey». In: *ACM Comput. Surv.* 48.1 (lug. 2015), 2:1–2:50. ISSN: 0360-0300. DOI: 10.1145/2767005. URL: http://doi.acm.org/10.1145/2767005.

[4] N. Gruschka e L. L. Iacono. «Vulnerable Cloud: SOAP Message Security Validation Revisited». In: *2009 IEEE International Conference on Web Services*. Lug. 2009, pp. 625–631. DOI: 10.1109/ICWS.2009.70.

[5] Sven Bugiel et al. «AmazonIA: When Elasticity Snaps Back». In: *Proceedings of the 18th ACM Conference on Computer and Communications Security*. CCS '11. Chicago, Illinois, USA: ACM, 2011, pp. 389–400. ISBN: 978-1-4503-0948-6. DOI: 10.1145/2046707.2046753. URL: http://doi.acm.org/10.1145/2046707.2046753.

[6] Thomas Ristenpart et al. «Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-party Compute Clouds». In: *Proceedings of the 16th ACM Conference on Computer and Communications Security*. CCS '09. Chicago, Illinois, USA: ACM, 2009, pp. 199–212. ISBN: 978-1-60558-894-0. DOI: 10.1145/1653662.1653687. URL: http://doi.acm.org/10.1145/1653662.1653687.

[7] M. Green. «The Threat in the Cloud». In: *IEEE Security Privacy* 11.1 (gen. 2013), pp. 86–89. ISSN: 1540-7993. DOI: 10.1109/MSP.2013.20.

[8] Huan Liu. «A New Form of DOS Attack in a Cloud and Its Avoidance Mechanism». In: *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop*. CCSW '10. Chicago, Illinois, USA: ACM, 2010, pp. 65–76. ISBN: 978-1-4503-0089-6. DOI: 10.1145/1866835.1866849. URL: http://doi.acm.org/10.1145/1866835.1866849.

[9] F. Rocha e M. Correia. «Lucy in the sky without diamonds: Stealing confidential data in the cloud». In: *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*. Giu. 2011, pp. 129–134. DOI: 10.1109/DSNW.2011.5958798.

[10] Sören Bleikertz et al. «Client-controlled Cryptography-as-a-service in the Cloud». In: *Proceedings of the 11th International Conference on Applied Cryptography and Network Security*. ACNS'13. Banff, AB, Canada: Springer-Verlag, 2013, pp. 19–36. ISBN: 978-3-642-38979-5. DOI: 10.1007/978-3-642-38980-1_2. URL: http://dx.doi.org/10.1007/978-3-642-38980-1_2.

[11] S. De Capitani di Vimercati et al. «Three-server swapping for access confidentiality». In: *IEEE Transactions on Cloud Computing* PP.99 (2015), pp. 1–1. ISSN: 2168-7161. DOI: 10.1109/TCC.2015.2449993.

[12] S. De Capitani di Vimercati et al. «Integrity for join queries in the cloud». In: *IEEE Transactions on Cloud Computing* 1.2 (lug. 2013), pp. 187–200. ISSN: 2168-7161. DOI: 10.1109/TCC.2013.18.

[13] S. A. Almulla e Chan Yeob Yeun. «Cloud computing security management». In: *2010 Second International Conference on Engineering System Management and Applications*. Mar. 2010, pp. 1–7.

[14] Hassan Takabi e James B.D. Joshi. «Policy Management as a Service: An Approach to Manage Policy Heterogeneity in Cloud Computing Environment». In: *45th Hawaii International Conference on System Sciences (HICSS-45)*. IEEE, 2012, pp. 5500–5508. URL: http://d-scholarship.pitt.edu/13526/.

[15] Philogene A. Boampong e Luay A. Wahsheh. «Different Facets of Security in the Cloud». In: *Proceedings of the 15th Communications and Networking Simulation Symposium*. CNS '12. Orlando, Florida: Society for Computer Simulation International, 2012, 5:1–5:7. ISBN: 978-1-61839-785-0. URL: http://dl.acm.org/citation.cfm?id=2331762.2331767.

[16] F. John Krautheim. «Private Virtual Infrastructure for Cloud Computing». In: *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing*. HotCloud'09. San Diego, California: USENIX Association, 2009. URL: http://dl.acm.org/citation.cfm?id=1855533.1855538.

[17] Stefan Berger et al. «vTPM: Virtualizing the Trusted Platform Module». In: *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*. USENIX-SS'06. Vancouver, B.C., Canada: USENIX Association, 2006. URL: http://dl.acm.org/citation.cfm?id=1267336.1267357.

[18] Michael Velten e Frederic Stumpf. «Secure and Privacy-Aware Multiplexing of Hardware-Protected TPM Integrity Measurements among Virtual Machines». In: *Information Security and Cryptology – ICISC 2012: 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*. A cura di Taekyoung Kwon, Mun-Kyu Lee e Daesung Kwon. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 324–336. ISBN: 978-3-642-37682-5. DOI: 10.1007/978-3-642-37682-5_23. URL: http://dx.doi.org/10.1007/978-3-642-37682-5_23.

[19] Jakub Szefer e Ruby B. Lee. «Hardware-Enhanced Security for Cloud». In: *Secure Cloud Computing*. Berlin: Springer, 2014, pp. 57–76. URL: http://link.springer.com/chapter/10.1007%2F978-1-4614-9278-8_3.

[20] C. A. Ardagna et al. «On the Management of Cloud Non-Functional Properties: The Cloud Transparency Toolkit». In: *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*. Mar. 2014, pp. 1–4. DOI: 10.1109/NTMS.2014.6814039.

[21] K.M. Goertzel et al. *Software Security Assurance: A State-of-the Art Report (SOAR)*. Information Assurance Technology Analysis Center, 2007. URL: https://books.google.it/books?id=xxHPMgEACAAJ.

[22] Iain MacNeil e Xiao Li. «"Comply or Explain": market discipline and non-compliance with the Combined Code». In: *Corporate Governance: An International Review* 14.5 (2006), pp. 486–496. URL: http://EconPapers.repec.org/RePEc:bla:corgov:v:14:y:2006:i:5:p:486-496.

[23] L. M. Riungu, O. Taipale e K. Smolander. «Research Issues for Software Testing in the Cloud». In: *2010 IEEE Second International Conference on Cloud Computing Technology and Science*. Nov. 2010, pp. 557–564. DOI: 10.1109/CloudCom.2010.58.

[24] Z. Chiba et al. «A survey of intrusion detection systems for cloud computing environment». In: *2016 International Conference on Engineering MIS (ICEMIS)*. Set. 2016, pp. 1–13. DOI: 10.1109/ICEMIS.2016.7745295.

[25] M. Ficco, L. Tasquier e R. Aversa. «Intrusion Detection in Cloud Computing». In: *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. Ott. 2013, pp. 276–283. DOI: 10.1109/3PGCIC.2013.47.

[26] E. et al. Damiani. *Cumulus. D2.3 Certification models v.2*. Seventh Framework Programme - CUMULUS FP7, 2014.

[27] Marco Anisetti et al. «A Test-based Security Certification Scheme for Web Services». In: *ACM Trans. Web* 7.2 (mag. 2013), 5:1–5:41. ISSN: 1559-1131. DOI: 10.1145/2460383.2460384. URL: http://doi.acm.org/10.1145/2460383.2460384.

[28] E. Damiani et al. «WS-Certificate». In: *2009 Congress on Services - I*. Lug. 2009, pp. 637–644. DOI: 10.1109/SERVICES-I.2009.132.

[29] M. Anisetti, C. Ardagna e E. Damiani. «2012». In: (Low-Cost Security Certification Scheme for Evolving Services).

[30] M. Anisetti, C. A. Ardagna e E. Damiani. «Security Certification of Composite Services: A Test-Based Approach». In: *2013 IEEE 20th International Conference on Web Services*. Giu. 2013, pp. 475–482. DOI: 10.1109/ICWS.2013.70.

[31] G. Spanoudakis, E. Damiani e A. Maña. «Certifying Services in Cloud: The Case for a Hybrid, Incremental and Multi-layer Approach». In: *2012 IEEE 14th International Symposium on High-Assurance Systems Engineering*. Ott. 2012, pp. 175–176. DOI: 10.1109/HASE.2012.16.

[32] N. S. Chauhan, A. Saxena e J. Murthy. «An Approach to Measure Security of Cloud Hosted Application». In: *2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. Ott. 2013, pp. 1–6. DOI: 10.1109/CCEM.2013.6684427.