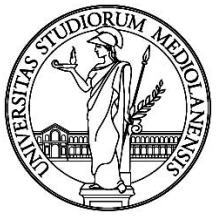

UNIVERSITÀ DEGLI STUDI DI MILANO

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 invece quella del NIST[1] nel documento *SP-800-145* 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 necessaria all'adozione di un servizio cloud.

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 virtualizza-

[1] National Institute of Standards and Technology

zione, risorse di rete e storage. Essendo il modello più di difficile gestione, è spesso amministrato tramite un orchestratore.

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.

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.

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. Queste, 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[2], garantendo la continuità di servizio riducendo i costi. Tuttavia esistono dei prerequisiti per garantire la

[2]Denial of Service

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[2]. Uno dei possibili approcci può essere la fornitura di metodologie di *security assurance* per la cloud transparency[3]. Per *security assurance* si intende la modalità per ottenere, con un certo livello di confidenza, 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[4].

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.[2]. Le vulnerabilità sono qui categorizzate in tre macro aree, in base alla superficie di attacco:

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 [2] sono:

- **Gruschka and Iacono, 2009**[], 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**[], 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[2].

Tra questi attacchi

-

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) [2].

Ciao sono patrizio

2.4 Tecniche di sicurezza per la cloud

2.5 Approcci per assurance, testing, monitoraggio e compliance

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] 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.

[3] 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.

[4] 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.