UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica Musicale

Mooncloud recommendation system

Relatore:
Claudio Agostino Ardagna
Correlatore:
Nome COGNOME

Tesi di Laurea di:
Andrea Michele Albonico
Matricola: 886667

Anno Accademico 2019/2020

Ringraziamenti

Andrea Michele Albonico

Prefazione

I sistemi di raccomandazione (*Recommendation System*) hanno avuto un forte sviluppo negli ultimi decenni e nascono proprio con lo scopo di identificare quegli oggetti (detti generalmente *item*) all'interno di un vasto mondo di informazioni che possono essere di nostro interesse e tanto maggiore è il grado di conoscenza dell'individuo e tanto più vengono ritenuti affidabili.
Il motivo di questo successo risiede nella riuscita integrazione di tali sistemi in applicazioni commerciali, soprattutto nel mondo dell'E-commerce e nel fatto che sono in grado di aiutare un utente a prendere una decisione che sia la scelta di un film per l'uscita con gli amici il sabato sera, di una playlist da ascoltare durante un viaggio in auto o in un momento di lettura, e via discorrendo.
MoonCloud è una piattaforma erogata come servizio che fornisce un meccanismo di *Security Governance* centralizzato. Garantisce il controllo della sicurezza informatica in modo semplice e intuitivo, attraverso attività di test e monitoraggio periodiche e programmate (*Security Assurance*). L'obbiettivo di questa tesi è stato quello di aggiungere, al già presente sistema per la scelta dei controlli all'interno delle attività di test, un sistema di raccomandazioni che possa consigliare all'utente delle possibili *evaluation* rispetto ai dati relativi al target indicato; in questo modo anche l'utente meno esperto può usufruire dei servizi offerti da MoonCloud in modo semplice e intuitivo.

La tesi è organizzata come segue:

Capitolo 1 – Introduzione a MoonCloud descrizione e funzionamento della piattaforma MoonCloud in ambito di Security Assurance.

Capitolo 2 – Tecnologie studi e analisi di soluzioni esistenti, studi delle tecnologie utilizzate nel seguito del lavoro.

Capitolo 3 – Descrizione delle attività svolte per conseguire gli obiettivi: Descrivere le attività svolte, riportando attivita, tempi, strumenti utilizzati, risultati conseguiti, problemi affrontati e modalita di risoluzione. Potranno essere qui descritte le attività anche dal punto di

vista strettamente tecnico, approfondendo le scelte effettuate, le motivazioni, le alternative prese in considerazione, l'uso o il possibile uso dei risultati del lavoro.

Capitolo 4 – Presentazione dei risultati e conclusioni] La presentazione dei risultati dovrebbe consistere in una descrizione tecnica dei risultati raggiunti,unitamente ad un commento critico e ad un'analisi della rispondenza agli obiettivi iniziali (si consiglia pertanto di motivare la rilevanza dei risultati e l'eventuale scostamento dagli obiettivi iniziali). La sezione relativa ai risultati dovrebbe infine contenere una sintesi critica e un giudizio sull'esperienza effettuata, che renda conto di aspetti positivi e negativi per il tirocinante e per l'ente ospitante, del valore formativo, professionale e umano, così via.

Indice

Elenco delle figure

Capitolo 1

Introduzione

In questo capitolo verrà descritto in modo più approfondito il funzionamento della piattaforma Moon Cloud e unitamente al motivo dell'implementazione della soluzione proposta.

1.1 MoonCloud overview

La diffusione di sistemi ICT (*Information and Communications Technology*) nella maggiorparte degli ambienti lavorativi e privati in termini di servizi offerti, automazione di processi e incremento delle performance. L'uso di questa tecnologia ha assunto importanza a partire dagli anni novanta come effetto del boom di Internet. Oggi le professionalità legate all'ICT crescono in numero e si evolvono per specificità, per operare in ambienti fortemente eterogenei ma sempre più interconnessi fra di loro come il cloud computing, i social newtwork, il marketing digitale, i sistemi IoT, la realtà virtuale, etc.

Gli immensi benefici del cloud in termini di flessibilità, consumo delle risorse e gestione semplificata, la rende la prima scelta per utenti e industrie per il deploy dei loro sistemi IT. Tuttavia il cloud computing solleva diverse problematiche legate alla mancanza di fiducia e trasparenza dove i clienti necessitano di avere delle garanzie sui servizi cloud ai quali si affidano; spesso i fornitori di servizi cloud non fornisco ai clienti le specifiche riguardanti le misure di sicurezza messe in atto.

Negli ultimi anni, sono state sviluppate tecniche e modi per rendere sicuri questi sistemi e proteggere i dati degli utenti, portando alla diffusione di approcci eterogenei che incrementaro la confusione negli utenti. Tecniche tradizionali di verifica della sicurezza basati su approcci di analisi statistica non sono più sufficenti e devono essere integrati con processi di raccolta di evidence da sistemi cloud in produzione e funzionamento. In generale il

cloud security definisce i modi (ad esempio criptazione, controllo degli accessi, etc.) per proteggere attivamente gli asset da minaccie interne ed esterne, e fornire un ambiente in cui i clienti possano affidarsi e interagire in totale sicurezza. Ma per rendere il cloud degno di fiducia e trasparente, sono state introdotte tecniche di *security assurance* le quali sono definite come il modo per ottenere la fiducia necessaria nelle infrastrutture e/o nelle applicazioni di dimostrare che siano garantite delle proprietà di sicurezza, e che operi normalmente anche se subisce attacchi; grazie alla raccolta e allo studio di queste evidence è possibile che venga accertata la validità e efficenza delle proprietà di sicurezza.

Il prezzo che paghiamo per i benefici di queste tecnologie è dato dall'incremento di violazioni di sicurezza, che oggigiorno preoccupa tutte le aziende e anche i loro clienti, con l'incremento del rischio di fallimento per i servizi più importanti dovuti a violazioni della privacy e al furto di dati.
Il mercato sta lentamente notando che non è l'inadequamento tecnologico dei sistemi di sicurezza che incrementa il rischio di furti di dati o delle violazioni di sicurezza; piuttosto, la mal configurazione e l'errata integrazione di questi sistemi nei processi di business. [2]

Per questo motivo anche se vengono usati i sistemi di sicurezza e di controllo migliori non è possibile garantire la sicurezza; ma è necessario implementare un processo continuo di diagnostica che verifica che i controlli siano configurati in modo corretto e il loro comportamento sia quello aspettato.

Il *Security Assesment* diventa allora un aspetto importante specialmente negli ambienti cloud e IoT. Questo processo deve essere portato avanti in modo continuo e olistico, per correlare le evidence raccolte da sempre maggiori meccanismi di protezione. [1]
Moon Cloud è una soluzione PaaS (Platform as a Service) che fornice una piattaforma B2B (Business To Business) innovativa per verifiche, diagnostiche e monitoraggio dell'adeguatezza dei sistemi ICT rispetto alle politiche di sicurezza, in modo continuo e su larga scala. Moon Cloud supporta una semplice ed efficente *ICT security governance*, dove le politiche di sicurezza possono essere definite dalle compagnie stesse (a partire da un semplice controllo sulle vulnerabilità a linee guida di sicurezza interna), da entità esterne, imposte da standard oppure da regolamentazioni nazionali/internazionali.
La sicurezza di un sistema o di un insieme di asset dipende solo parzialmente dalla forza dei singoli meccanismi di protezione isolati l'uno dall'altro; infatti, dipende anche dall'abilità di questi meccanismi di lavorare continuamente in sinergia per provvedere a una protezione olistica. In più, quando i sistemi cloud e i servizi IoT sono coinvolti, le dinamiche di questi servizi e la loro rapida evoluzione rende il controllo dei processi all'interno dell'azienda e le politiche di sicurezza più complesse e prone ad errori.

I requisiti ad alto livello fondamentali per poter garantire le Security Assurance sono:

sistema olistico è richiesta una visione globale e pulita dello stautus dei sistemi di sicurezza; inoltre è cruciale distribuire lo sforzo degli specialisti in sicurezza per migliorare il processo e le politiche messe in atto. Si parte da delle valutazioni fatte manualmente a quella semi-automatiche che ispezionano i meccanismi di sicurezza.

monitoraggio continuo ed efficiente è necessario un controllo continuo che valuti l'efficenza dei sistemi di sicurezza per ridurre l'impatto dell'errore umano, soprattutto dal punto di vista organizzativo. La mancata configurazione dovuta al cambiamento dell'ambiente, la coesistenza di componenenti in conflitto: sono scenari che richiedono un monitoraggio e un aggiornamento continuo.

singolo punto managment avere un solo punto in cui gestire tutti gli aspetti relativi alla sicurezza, permette di avere sotto controllo le politiche di sicurezza. Inoltre disporre di un inventario degli asset da proteggere, così da poter conoscere quali protezioni applicare.

reazioni rapide a incidenti di sicurezza spesso la reazione ad incidenti di sicurezza è ritardata da due fattori: il tempo richiesto per rilevare l'incidente e il tempo per analizzare il motivo dell'accaduto.

Moon Cloud è basato su una tecnica di Security Assurance garantendo che tutte le attività aziendali si compiano seguendo i requisiti prestabiliti da appropriate politiche e procedure.

Una *Security Compliance Evaluation* è il processo di verifica a cui un target viene sottoposto e il cui risultato deve soddisfare i requisiti richiesti da standard e politiche. A partire da questi processi di verifica, che devono a loro volta essere affidabili, si ottendo delle evidence; queste ultime possono essere raccolte monitorando l'attività del target oppure, come già menzionato, sottoponendo il target a scenari critici o di testing. In particolare, una Security Compliance Evaluation è un processo che verifica l'uniformità di un certo target a una o più politiche attraverso una serie di controlli che a seconda dvalore booleano associato ad ogni controllo viene prodotto un valore booleano per le politiche.

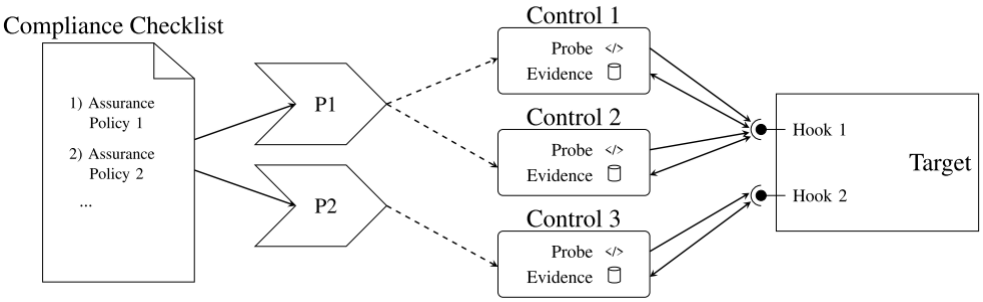

Figura 1.1: Security Compliance Evaluation

1.2 Processo di Evaluation

Moon Cloud implementa il processo di Security Compliance Evaluation in Figura 1 usando controlli di monitoraggio o di test personalizzabile. Inoltre garantisce, oltre a tutti i requisisti ad alto livello elencati prima, anche i seguenti:

Moon Cloud è una piattaforma cloud centralizzata presentando una visione olistica dello stato di sicurezza di un dato sistema.

Moon Cloud implementa un sistema di Assurance Evidence-based continuo, implementato come processo di Compliance, basato su politiche custom o standard.

Moon Cloud è offerto come un servizio - PaaS, dove le attività di evaluation possono essere facilmente ed efficentemente configurate su un target asset, senza l'intervento dell'uomo.

Moon Cloud permette di schedulare delle ispezioni automatiche, grazie all'inventario di asset protetto.

Moon Cloud evaluation engine può ispezionare dall'interno, gestendo delle minaccie interne; permettendo anche reazioni rapide a incidenti di sicurezza e veloci rimedi, grazie alla raccolta continua di evidence.

L'architettura di Moon Cloud è costituita da un'Assurance Manager che gestisce i processi di Evaluation attraverso un set di *Execution Cluster*; ognuno dei quali gestisce ed esegue un set di probe collezionando le evidence necessarie per le evaluation. Tutte le attività di collezione sono eseguite dal probe. Ogni probe è uno script di python fornito come una sigola immagine di Docker, che viene inizializzata quando è triggerata una evaluation ed è

distrutta quando il processo di evaluation è terminato.
Accedendo alla piattaforma di Moon Cloud, l'utente può definire le proprie politiche di sicurezza e attività di evaluation come espressioni booleane di controlli di sicurezza e altre politiche predefinite. Una volta che una politica viene definita, l'utente può decidere quando schedulare l'evaluation; e nel momento in cui un processo di evaluation viene inizializzato, tutti i controlli vengono eseguiti e i risultati dell'espressioni booleane vengono memorizzati e restituiti all'utente. A questo punto l'utente può accedere a questi risultati a diversi gradi di precisione: una visione sommaria e generale di tutte le politche implentate e dello stato generale del sistema di sicurezza, al risultato di una specifica politica oppure alle evidence raccolte per una evaluation.

Capitolo 2

Tecnologie

In questo capitolo verranno descritte le attività preliminari per la realizzazione di questo progetto, le tecnologie utilizzate unitamente alle motivazioni legate all'uso di questi sistemi rispetto ad altri.

2.1 Strutture dati gerarchiche

Le tabelle di un database relazione non sono gerarchiche (come nel XML), ma sono delle semplici liste piatte. I dati gerarchici sono constituiti da relazioni padre-figlio che non possono essere rappresentate in modo naturale nelle tabelle dei database relazionali. In questo caso, i dati gerarchici sono una collezione di informazioni dove ogni item ha un solo padre e nessuno o più figli (ad eccezione del nodo radice che non ha un nodo padre); questo genere di rappresentazione delle informazioni può essere trovato in diversi ambiti di applicazione di un database, incluse discussioni su forum e mailing list, grafici di organizzazione di un business, categorie per gestire contenuti e categorie di prodotti.

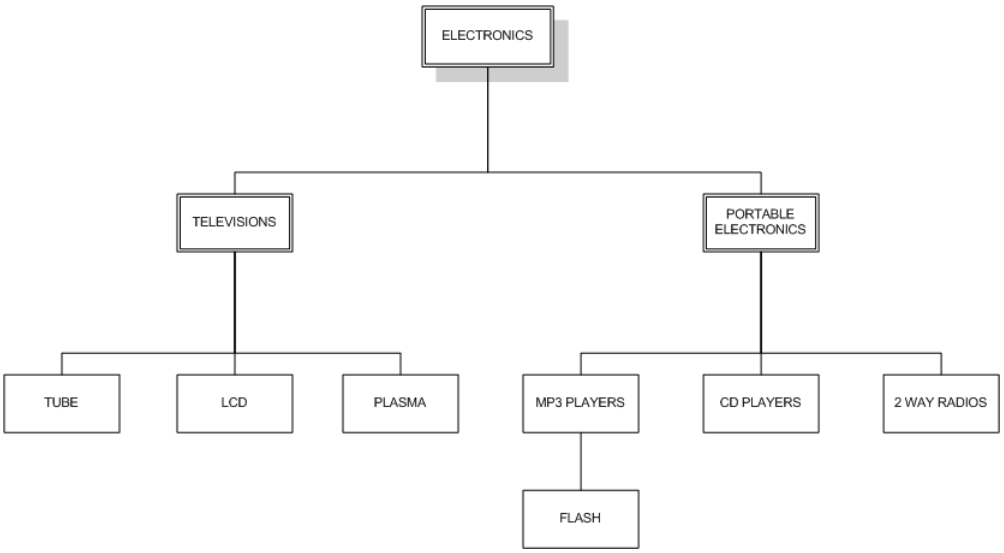

Figura 2.1: Esempio di una gestione di dati in modo gerarchico

Ci sono differenti modelli per poter gestire dati in modo gerarchico, i più importanti che sono stati presi in considerazione sono i seguenti:

2.1.1 The adjacency list model

Il primo approccio, e quello di più semplice implementazione, qui descritto è chiamato *'adjacency list model* o metodo ricorsivo; è definito tale perchè per funzionare necessita solo di una funzione che itera per tutto l'albero.

In questo modello, ogni item (nodo dell'albero) nella tablla contiene un puntatore al suo item padre; invece il nodo radice avrà un puntatore a un valore NULL per l'item padre.

```
CREATE TABLE category(
        category_id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(20) NOT NULL,
        parent INT DEFAULT NULL
);

INSERT INTO category VALUES(1,'ELECTRONICS',NULL),(2,'TELEVISIONS',1),(3,'TUBE',2),
        (4,'LCD',2),(5,'PLASMA',2),(6,'PORTABLE ELECTRONICS',1),(7,'MP3 PLAYERS',6),
(8,'FLASH',7),
        (9,'CD PLAYERS',6),(10,'2 WAY RADIOS',6);

SELECT * FROM category ORDER BY category_id;
+-------------+----------------------+--------+
| category_id | name                 | parent |
+-------------+----------------------+--------+
|           1 | ELECTRONICS          |   NULL |
|           2 | TELEVISIONS          |      1 |
|           3 | TUBE                 |      2 |
|           4 | LCD                  |      2 |
|           5 | PLASMA               |      2 |
|           6 | PORTABLE ELECTRONICS |      1 |
|           7 | MP3 PLAYERS          |      6 |
|           8 | FLASH                |      7 |
|           9 | CD PLAYERS           |      6 |
|          10 | 2 WAY RADIOS         |      6 |
+-------------+----------------------+--------+
```

Figura 2.2: Esempio di una tabella per gestire dati in modo gerarchico secondo l' adjacency list model

Il vantaggio di usare questo modello sta nella sua semplicità di costruzione sopratutto a livello di codice client-side, e di restituzione dei figli di un nodo. Mentre diventa problematico se si lavora in puro SQL e nella maggior parte dei linguaggi di programmazione, è lento e poco efficente, perchè è necessaria una query per ogni nodo dell'albero, e visto che ogni query impiega un certo periodo di tempo, questo rende la funzione molto lente quando si lavora con alberi di grandi dimensioni. Inoltre molti linguaggi non sono ottimizzati per

funzioni ricorsive. Per ogni nodo, la funzione inizia una nuova istanza di se stessa, ogni istanza occupa una porzione di memoria e impiega un certo tempo per inizializzarsi, e più grande è l'albero e più questo processo sarà portato a termine in maggior tempo.

2.1.2 The Nested set model

Il secondo approccio che viene proposto è il *Nested set model*, che permette di osservare la gerarchia in un modo diverso, non come nodi e linee, ma come container innestati.

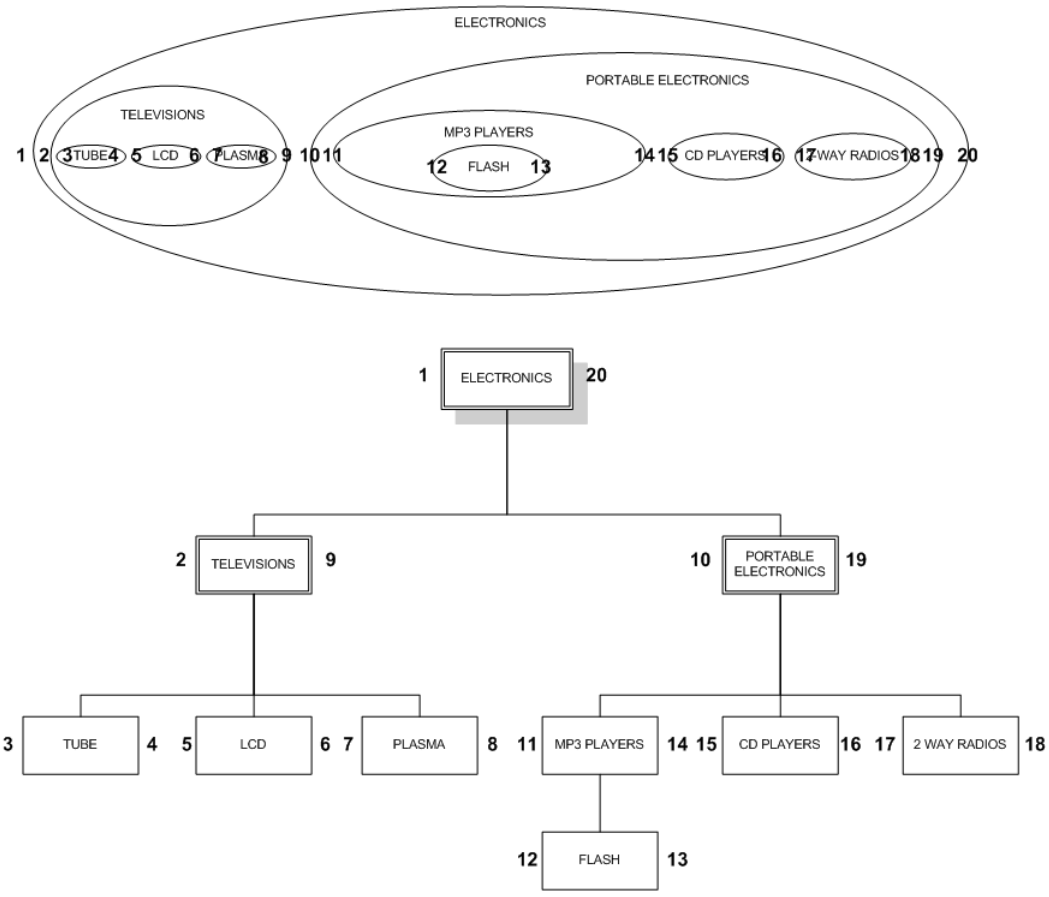

Figura 2.3: Esempio di una gestione di dati in modo gerarchico secondo il Nested set model

La gerarchia dei dati viene rappresentata nella tabella attraverso l'uso degli attributi 'left' e 'right' per rappresentare l'annidamento dei nodi (il nome delle colonne: left e right, hanno significati speciali in SQL; per questo motivo si identificano questi campi con i nomi 'lft' e 'rght'). Ogni nodo dell'albero viene visitato due volte, assegnando i valori in ordine di visita, e in entrambe le visite. Quindi vengono associati ad ogni nodo due numeri, memorizzato come due attributi. I valori di left e right sono determinati come segue: si inizia a numerare a partire dal lato più a sinistra di ogni nodo

e si continua verso destra. Lavorando con un albero, si parte da sinistra e si continua verso destra, un livello alla volta, scendendo per ogni nodo i suoi figli, assegnando i valori al campo left, prima di assegnare un valore al campo right, e successivamente si continua verso destra. Questo approccio è chiamato Modified preorder tree traversal algorithm.

A prima vista questo approccio può sembrare più complicato da comprendere rispetto all'adjacency list model, ma quest'utlimo metodo è molto più veloce quando si vuole recuperare i nodi, visto che basta una query, mentre più lento per operazioni di aggiornamento e cancellazione dei nodi; in quest ultimo il grado di complicatezza dell'operazione è determinato dal nodo che si vuole cancellare, a partire dal caso più semplice, il nodo foglia (nodo senza figli) fino al caso più complicato, quando si vuole cancellare il nodo radice.

```
CREATE TABLE nested_category (
        category_id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(20) NOT NULL,
        lft INT NOT NULL,
        rgt INT NOT NULL
);

INSERT INTO nested_category VALUES(1,'ELECTRONICS',1,20),(2,'TELEVISIONS',2,9),(3,'TUBE',3,4),
 (4,'LCD',5,6),(5,'PLASMA',7,8),(6,'PORTABLE ELECTRONICS',10,19),(7,'MP3 PLAYERS',11,14),
(8,'FLASH',12,13),
 (9,'CD PLAYERS',15,16),(10,'2 WAY RADIOS',17,18);

SELECT * FROM nested_category ORDER BY category_id;

+-------------+----------------------+-----+-----+
| category_id | name                 | lft | rgt |
+-------------+----------------------+-----+-----+
|           1 | ELECTRONICS          |   1 |  20 |
|           2 | TELEVISIONS          |   2 |   9 |
|           3 | TUBE                 |   3 |   4 |
|           4 | LCD                  |   5 |   6 |
|           5 | PLASMA               |   7 |   8 |
|           6 | PORTABLE ELECTRONICS |  10 |  19 |
|           7 | MP3 PLAYERS          |  11 |  14 |
|           8 | FLASH                |  12 |  13 |
|           9 | CD PLAYERS           |  15 |  16 |
|          10 | 2 WAY RADIOS         |  17 |  18 |
```

Figura 2.4: Esempio di una tabelle per la gestione di dati in modo gerarchico secondo il Nested set model

2.2 Sistemi di raccomandazione

Capitolo 3

Sistemi di raccomandazione

Capitolo 4

Descrizione approfondita del progetto

Capitolo 5

Conclusioni

Bibliografia

[1] M. Anisetti et al. «A semi-automatic and trustworthy scheme for continuous cloud service certification». In: *IEEE TRANSACTIONS ON SERVICES COMPUTING* (2017). DOI: 10.1109/TSC.2017.2657505.

[2] M. Anisetti et al. «Moon Cloud: A Cloud Platform for ICT Security Governance». In: (dic. 2018), pp. 1–7. DOI: 10.1109/GLOCOM.2018.8647247.