Lezione 13 modulo 2

I dati all'interno del sistema informativo sono chiaramente la risorsa che vogliamo proteggere e per cui vogliamo garantire delle proprietà di sicurezza, in particolare ci occuperemo qui dell'accesso ai dati per quanto riguarda la proprietà soprattutto di riservatezza dei dati e quindi la capacità di poter leggere certi dati da parte degli utenti che vengono autorizzati a farlo e, in generale, quindi la proprietà è la proprietà di riservatezza, ma vogliamo anche proteggere il sistema per quanto riguarda possibili inserimenti dei dati, quindi quello che vogliamo proteggere è l'integrità dei dati e quindi vorremo definire dei meccanismi per, appunto, definire quali sono le regole per l'accesso ai dati da parte dei vari utenti. Il meccanismo generale che possiamo avere per l'accesso ai dati è quello di filtrare delle richieste d'accesso al sistema, attraverso delle procedure di controllo e poi permettere l'accesso oppure negare l'accesso. Abbiamo detto, questo avverrà sulla base di regole che vengono stabilite e queste regole saranno definite sulla base di politiche di sicurezza che verranno definite all'interno di un'organizzazione. Quello che noi andiamo adesso a vedere è la definizione di quali sono gli elementi che compongono le regole, quali sono le tipologie di politiche e, poi, come possiamo appunto controllare l'accesso utilizzando diversi tipi di tecniche. Come primo elemento vediamo gli elementi delle regole: noi vedremo che le nostre regole potranno essere più o meno articolate, questo lo discuteremo nel corso di questa lezione. Innanzitutto noi vogliamo considerare il caso in cui abbiamo delle autorizzazioni ad accedere ai dati, vediamo innanzitutto basate sul concetto di ownership dei dati, quindi proprietà dei dati. Noi vedremo che questo non sarà l'unico meccanismo possibile; in questo caso abbiamo dei soggetti a cui verranno dati dei diritti sulla base, abbiamo detto, in genere dei profili, i profili possono essere profili di gruppo, quindi potranno avere ad esempio un 'gruppo ID', oppure profili legati a un certo gruppo di applicazioni, quindi 'application ID' o anche al ruolo che gli utenti hanno all'interno dell'organizzazione, quindi 'Role ID', quindi il soggetto non viene visto in genere come solo la singola persona ma anche come appartenente all'organizzazione con una certa tipologia di profilo associato. Ovviamente lo stesso soggetto potrebbe anche avere un'appartenenza a più ruoli, a più gruppi, un accesso a più applicazioni. Questi soggetti accedono a degli oggetti, quindi quello che noi vorremmo definire sono delle regole che appunto ci diranno quali sono i diritti d'accesso che i vari soggetti hanno sugli oggetti. Questi diritti, in generale, trattandosi di dati, di solito sono quelli che sono relativi all' utilizzo del dato, quindi la creazione del dato, in genere si usa l'acronimo 'CRUD' per dire creazione, lettura (Read), (Update) aggiornamento oppure (delete), la cancellazione. Quindi noi vorremmo dire per ciascun dato quali sono i diritti, quindi quali sono le possibili operazioni che l'utente può fare sul dato. Ovviamente quando stiamo parlando di oggetti, dipenderà dal tipo di sistema che stiamo utilizzando. Tipicamente noi andremo a parlare di oggetti che sono dei dati e più in particolare quando parleremo delle realizzazioni, parleremo di tabelle relazionali, quindi vedremo come dare questi diritti all'interno di basi di dati relazionali. Un altro elemento che abbiamo visto sono le 'politiche': le politiche sono quelle che ci dicono come definire le regole e, in particolare, noi parleremo di due tipi di politiche: quelle che sono 'di tipo aperto', quindi io andrò sostanzialmente a definire quello che viene vietato e quelle 'di tipo chiuso' e quindi in questo caso definisco esplicitamente le regole che ci dicono solo gli accessi consentiti, per cui in un sistema aperto andrò a vedere se nulla impedisce di accedere a un certo dato, in un sistema chiuso andrò a verificare che effettivamente ci sia una regola che dia questo permesso. Vediamo quindi adesso come possono essere definite queste regole: abbiamo innanzitutto delle regole che possono essere definite 'a tre componenti', sostanzialmente legherò soggetto, l'oggetto e il diritto d'accesso che abbiamo definito. Spesso questi componenti in realtà vedremo potranno essere di più perché voglio specificare più in dettaglio alcune caratteristiche dell'accesso, ad esempio una prima estensione è mettere dei vincoli, quindi quando ho quattro componenti, avrò anche dei vincoli o 'constraint' che verranno aggiunti, che definiscono gli ambiti permessi di accesso ai dati. Questo lo possiamo vedere in un esempio che è un esempio riassuntivo di come può essere descritto l'accesso a una tabella relazionale chiamata 'Employee', impiegati, con dati anagrafici sugli impiegati (nome, indirizzo, età) e abbiamo la possibilità di fare appunto degli

accessi da parte di soggetti definiti qua come ruoli, quindi manager, impiegato e come diritti abbiamo i soliti diritti crowd. Vediamo ad esempio nella figura, che per gli impiegati di tipo manager, abbiamo dei vincoli diversi per quanto riguarda gli accessi ai dati. Per impiegati di cui il manager è manager, avrà tutti i diritti d'accesso, mentre invece gli altri manager avranno solo i diritti di lettura per gli impiegati di cui non sono manager. Per quanto riguarda un impiegato vediamo qua due regole: il fatto che posso andare a leggere e anche aggiornare i dati anagrafici dell'impiegato stesso, quindi quando l'impiegato è quello che sta aggiornando i suoi dati, oppure posso andare a leggere i dati che sono nome, età e indirizzo in cui il manager del dato che si va a leggere è il manager dell'impiegato stesso. Quindi, sostanzialmente all'interno ad esempio del proprio dipartimento. Per quanto riguarda le politiche, abbiamo detto che possiamo avere politiche di vario tipo, ma anche un'altra cosa che va precisata è che le politiche possono distinguersi a seconda del modo con cui trattano i diritti e gli accessi. In particolare abbiamo due tipologie: quelle che vengono chiamate 'discrezionali', si parlerà di DAC o ''Discretionary Access Control', in cui sostanzialmente si segue la tipologia di controllo degli accessi di cui abbiamo parlato finora quando abbiamo detto che diamo dei diritti e quindi abbiamo la capacità da parte di un proprietario del dato di dare questi diritti direttamente oppure di delegare qualcuno a dare questi diritti. C'è un altro tipo però di politica che si può utilizzare che è quella che viene chiamata 'mandatoria': si parlerà di MAC quando avremo un controllo degli accessi appunto di questo tipo. Perché abbiamo diversi tipi di politica? Il problema che noi possiamo avere nel caso DAC è che i dati che vengono considerati confidenziali, ad esempio all'interno di un ruolo oppure all'interno di un'associazione fra un certo soggetto e un oggetto, vengono definiti certi livelli, possono essere inseriti all'interno di elaborazione dei dati e si possono creare dei flussi di dati che sono anche non previsti all'interno del sistema, quindi c'è il rischio di creare dei flussi di dati non autorizzati. Abbiamo visto, ad esempio, che uno stesso utente può assumere all'interno del profilo associato una serie di ruoli, oppure appartenere a più gruppi e quindi avere visibilità diverse di dati. Nel caso in cui io ho utenti che sono autorizzati a vedere certi dati, secondo profili diversi, uno stesso utente potrebbe vedere i dati a diversi livelli di confidenzialità e quindi poter fare delle operazioni diverse. Questo può creare dei flussi non desiderati all'interno di una certa applicazione, anche inavvertitamente possiamo trasferire dei dati che dovrebbero essere confidenziali per altri profili. Per evitare questo, l'approccio mandatorio sostanzialmente si basa su un concetto che non è quello di assegnare un diritto ad un certo utente da parte del proprietario del dato su un certo oggetto, ma su un concetto di 'livello di classificazione'. Quando parliamo di classificazione definiremo, appunto, dei livelli e questi livelli potranno essere applicati sia ai soggetti che anche agli oggetti, quindi ai dati. Ad esempio possiamo avere dei livelli che vanno da 'top secret', a 'secret' e 'confidential' e poi a 'unclassified', quando tutti i soggetti possono vedere tutti gli oggetti. Quello che è importante nel caso del MAC è il fatto che questa classificazione si applica sia ai soggetti, che agli oggetti. Nel caso del MAC, quando andiamo a definire le regole, saranno delle regole che saranno dipendenti da questa classificazione e non saranno assegnati in modo discrezionale come abbiamo visto prima nel DAC, quindi sono due regole: una relativa alla lettura, 'No read up' e una relativa alla scrittura, che chiameremo 'No write down'. Quando diciamo che non vogliamo leggere a un livello superiore è quello che può essere intuitivo in una classificazione di questo tipo: una persona di livello unclassified non potrà leggere documenti top secret, quindi si applica all'operazione di lettura e noi diciamo che il livello del soggetto dovrà essere maggiore o uguale al livello dell'oggetto che viene letto. Nel caso della scrittura, noi vogliamo consentire anche a persone di livello inferiore di comunicare informazioni riservate, ad esempio superiore, quindi possiamo scrivere informazioni più confidenziali del nostro livello. Quindi, il nostro livello del soggetto, deve essere minore uguale al livello dell'oggetto. Quello che voglio sottolineare qui, è che siccome l'obiettivo del MAC è evitare flussi fra i livelli diversi non intenzionali, io non posso andare a scrivere degli oggetti di livello inferiore al mio livello di segretezza, come soggetto. Quindi abbiamo una regola, appunto, che viene sintetizzata in questa formula, che mi dice che posso scrivere oggetti al mio livello o a livello superiore ma non a livello inferiore. Su queste regole poi si baserà appunto l'accesso ai dati, quindi io non andrò a definire delle regole specifiche per la coppia soggetto- oggetto, come abbiamo fatto nel DAC, ma abbiamo delle regole fisse che valgono in tutti i casi. Quindi l'approccio MAC si

basa sulle classificazioni, in genere queste regole di classificazione potranno anche avere degli ambiti di sicurezza e quindi, ad esempio qua, vediamo che abbiamo definito due tipi di domini di sicurezza, uno nucleare, uno che è Intelligence e quindi noi possiamo associare delle gerarchie, appunto, che ci dicono a partire dalla gerarchia base che abbiamo visto prima qual è la classificazione di un oggetto, possiamo definire le gerarchie che ci dicono che un 'Confidential Nuclear' è inferiore a un 'Secret Nuclear', che è inferiore a un 'Secret Nuclear Intelligence' e un 'Confidential Nuclear' è di livello inferiore a un 'Secret nuclear Intelligence'. Quindi noi andremo a utilizzare queste classificazioni, queste gerarchie di classificazioni per associare a soggetti-oggetti la propria classificazione e sulla base di questo potremo applicare le regole appunto che abbiamo visto prima. Vediamo ora come queste politiche e appunto questo concetto di regola che abbiamo visto nei due casi DAC e MAC, vengono implementati su dei BDMS relazionali. Cominciamo a partire dal DAC e poi vedremo il MAC. Per quanto riguarda il DAC, supponiamo di avere delle tabelle, lo facciamo vedere con un esempio, in cui ad esempio nella tabella impiegato, abbiamo dei dati sull'impiegato e, abbiamo detto, possiamo definire ad esempio dei ruoli, possiamo definire dei ruoli che ci dicono che il capo del personale potrà vedere tutti i dati, mentre invece il fattorino che dovrà fare le consegne dovrà poter identificare dove sono le persone, quindi vedrà il codice impiegato, nome e reparto dove dovrà effettuare le consegne. Quindi, questo è il concetto del DAC: io assocerò a diversi ruoli la possibilità di vedere dei dati in modo diverso. Nel momento in cui noi vogliamo fare questo sistema relazionale, lo potremo fare associando con delle istruzioni SQL dei diritti d'accesso che ci daranno questa possibilità. Questo lo faremo con una istruzione, l'istruzione GRANT, che ci dirà quali sono i diritti d'accesso dati a un certo soggetto, oppure a un certo ruolo. Negli esempi che vedremo successivamente, saranno associati a dei soggetti, quindi data la relazione EMP definita con i suoi attributi noi scriveremo GRANT, poi dirò quale è l'operazione, che appunto sarà una operazione di inserimento, cancellazione o selezione, se vogliamo fare la lettura, quindi avremo qua la lista delle operazioni ammissibili su una certa tabella, quindi 'ON EMP' e poi diremo a chi, quindi avremo 'ON EMP TO John' per associare a John il diritto di inserire e anche leggere i dati su una tabella EMP. Ad esempio, abbiamo visto che però in alcuni casi noi vogliamo anche mettere dei vincoli aggiuntivi nell'accesso ai dati, ad esempio vogliamo vedere solo gli impiegati del reparto 50 e vogliamo far vedere solo gli attributi CEmp, nome e reparto. In questo caso, a partire dalla stessa relazione, possiamo utilizzare il concetto di vista che è già stato definito in SQL, quindi possiamo creare una vista, che ci dice che selezioniamo l'attributo CEmp, nome e reparto da EMP con il reparto uguale a 50 e poi usiamo di nuovo una GRANT con SELECT come operazione, per poter visualizzare le informazioni di lettura sulla vista, quindi, quando abbiamo la clausola ON possiamo avere sia una relazione che una vista e sempre abbiamo il soggetto, TO John, per dire a chi. In questo modo, possiamo realizzare accessi selettivi a partire dalle relazioni. Un altro tipo di vincolo che noi possiamo volere è il fatto di restringere non solo a una vista, ma anche restringere solo ad alcuni valori all'interno della relazione. Qui vediamo un esempio di GRANT selettivo, una GRANT SELECT, un EMP TO John, WHERE stipendio minore di 1.500, quindi voglio far vedere solo quelle tuple in cui il valore stipendio è inferiore a 1.500. Nel momento in cui ho messo questo vincolo, quando andrò a eseguire una query di questo tipo, nel caso io sia impiegato John e scrivo SELECT* FROM EMP, io vedrò non tutti i dati, ma solo quelli che rispettano la clausola del diritto d'accesso che è stata definita e quindi in realtà la mia query verrà trasformata da SELECT* FROM EMP a SELECT* FROM EMP WHERE Stipendio è minore di 1500. In questo modo, noi abbiamo fatto una restituzione quindi di tipo selettivo sulle tuple che vengono utilizzate da un certo utente, John, con un diritto di selezione su EMP ristretto con una clausola WHERE. Un altro tema che è importante è che la nostra istruzione GRANT in realtà viene associata a dei dati e dà dei diritti d'accesso a degli utenti. Chi potrà utilizzare l'istruzione GRANT? In teoria, il proprietario dei dati, in generale quello che noi abbiamo all'interno di un database è che il proprietario di tutti i dati è il 'Database Administrator', che potrà delegare la possibilità di dare eventualmente altri diritti d'accesso ad altri utenti. Quindi la nostra istruzione GRANT potrà essere associata a un'ulteriore clausola che viene chiamata 'With Grant Option', che dice che quanto è stato dato come diritto può essere trasferito ad altri. Seguiamo questa clausola, quindi noi possiamo avere che un certo diritto dato ad A viene dato a B e B a sua volta lo può dare a C. Abbiamo qui

un tema di propagazione dei diritti; ovviamente per lo stesso dato abbiamo detto che possiamo avere diversi motivi per accedere a un certo dato e potrei avere questo diritto, ad esempio per C, dato anche da un altro, supponiamo che il nostro A sia il database administrator da un altro utente D che è stato autorizzato dal database administrator. Ovviamente, rispetto all'operazione GRANT, c'è anche l'operazione inversa, che è l'operazione di revoca, REVOC, che abbiamo per revocare i diritti d'accesso. Cosa succede quando A revoca l'accesso a B? Si pone il tema del fatto che io posso avere appunto una necessità di revocare a cascata, perché se A ha dato un diritto a B e B l'ha dato a C, può essere che io voglia revocare anche il diritto a C, quindi una delle regole che potrei avere è revocare anche il diritto a C. Però se C ha ricevuto lo stesso diritto anche da B, continuerà ad averlo, ovviamente questo si presta a delle situazioni anche complesse, ad esempio cosa facciamo se C ha anche dato il suo diritto a B? Questo può essere connesso appunto all'utilizzo del dato nell'ambito di alcune attività da parte di C e di B ed essere possibile, oppure potrebbe essere considerato negato nel momento in cui A dice che B non può più accedere al dato, allora se ho avuto anche un diritto dato da C a B, voglio cancellare questo dato. Questo per dire che il tema è un tema di propagazione dei diritti e anche propagazione delle revoche, che può essere anche complesso perché questo grafo può avere cicli. Non è un albero, quindi abbiamo questo stesso diritto che può arrivare a un utente attraverso percorsi diversi e quindi si pongono diverse possibilità. In genere vengono utilizzate delle possibilità diverse, quindi delle politiche diverse rispetto alla propagazione. Quindi quando parliamo di propagazione degli accessi, arriviamo a definire delle regole che possono essere anche a sei componenti, in cui o chi dà il diritto, il GRANTOR o il GRANTEE, quello che riceve il diritto, l'oggetto, il diritto, l'accesso, RIGHT, i vincoli che è quanto abbiamo visto sostanzialmente finora. A questo aggiungiamo i 'Propagation constraint': noi sostanzialmente possiamo dare dei vincoli diversi che regolano in modo diverso il modo in cui un certo diritto si può propagare quando è stato dato a un certo utente. Una prima cosa che si può mettere è che non abbiamo nessuna propagazione; questo di solito è di default: se non consento di propagare il diritto esplicitamente questo non può essere trasferito ad altri, oppure la 'propagazione illimitata': l'utente una volta ricevuto il diritto può darlo ad altri senza limitazioni, l'ultima possibilità è mettere ovviamente dei limiti, quindi i limiti, che potranno essere limiti ad esempio di lunghezza della catena, oppure i limiti temporali che mi consentiranno con delle regole ulteriori di definire quali sono i limiti di propagazione per il diritto che viene dato all'interno di una certa istruzione GRANT che sto definendo. Vediamo adesso come possiamo realizzare un sistema relazionale tipo MAC, quindi basato sulla classificazione degli oggetti e poi avremo le nostre regole che guideranno la lettura sulla base della classificazione dei soggetti, la lettura e ovviamente anche la scrittura. Sostanzialmente, in una tabella relazionale noi tipicamente avremo una serie di attributi, quello che andrò a fare è associare ad ogni attributo la classificazione dell'attributo, quindi avremo l'attributo A1 con la classificazione 1, attributo An con la classificazione N, inoltre assocerò anche la classificazione all'intera tupla, quindi avremo una classificazione di tupla che verrà inserita. Vediamo un esempio: il nostro esempio della relazione impiegati, in cui noi abbiamo una classificazione, prendiamo ad esempio la prima tupla che è quella con nome Bob ed è tutta di livello Secret, quindi ogni attributo è di livello Secret, ma anche l'intera tupla è di livello Secret. Vediamo un'altra tupla in cui abbiamo Ann del dipartimento 2, che è vista a livello Secret ma il suo salario e l'intera tupla, la visibilità dell'intera tupla, sono di livello Top secret. L'ultimo caso invece è di nuovo un caso omogeneo, solo che è tutto di livello Top Secret. Cosa succede se un utente di livello Secret vede questa tupla? Allora questa è la nostra relazione come la vedevamo prima, il livello di tipo Secret non avrà la visibilità sul salario di Ann che è stato dichiarato top secret e potrà vedere, invece Bob non vedrà la tupla di Sam e vedrà solo alcune informazioni su Ann. Ovviamente, questo può essere un problema, nel senso che questo rivela che Ann ha un salario di una classificazione più elevata, in quanto un significato nullo in questo caso, potrebbe essere quello più che una mancanza del salario, nel fatto che salario non si voglia dichiarare. Un' altra cosa che può avvenire in sistemi di questo tipo è che si possano inserire valori in genere diversi a seconda della visibilità che si ha sull'informazione e quindi io posso inserire, come vi ricorderete, dei dati del mio livello in scrittura oppure di livello superiore, quindi io posso inserire un dato ad esempio qua di livello Secret, sempre di livello Secret, non vedo che c'è un livello Top secret e quindi

cosa avverrà dal punto di vista della rappresentazione dei dati? Potrà avvenire, quello che vediamo in questa figura: che abbiamo due tuple per Ann, in cui avremo uno stipendio che è visibile a livello Secret, ad esempio 15, mentre a livello Top Secret ha lo stipendio 20. Quindi, a seconda del visualizzatore, avrò una diversa informazione relativa allo stipendio di Ann. Questo viene poi realizzato con quella che viene chiamata la 'Polinstanziazione': verranno create immagini diverse di questo database a seconda della classificazione degli utenti nei livelli che sono stati definiti e che quindi darà loro la possibilità di fare operazioni di lettura e scrittura secondo le regole relative al proprio livello. Chiudiamo questa carrellata sull'accesso ai dati relazionali, andando a segnalare un altro problema di sicurezza che si può verificare, che viene chiamato 'SQL injection'. Il problema che ci può essere è che spesso le query che vengono utilizzate all'interno di un applicativo sono costruite dinamicamente dall'applicativo nell'esecuzione, quindi raccogliendo delle informazioni. Ad esempio una query di questo tipo che mi va a verificare se un utente ha le credenziali corrette per l'accesso al sistema, utilizza una tabella users e va a controllare se c'è una tupla all'interno della tabella che contenga le credenziali fornite dell'utente. Quindi, l'utente inserirà l'input username, per input password, che vengono utilizzate per comporre questa query, quindi darò l'accesso se effettivamente troverò uno username e una password all'interno della tabella. Questa query viene costruita appunto da uno schema che è quello che mi dà lo schema generale della query, la SELECT oppure lo username '=' e il valore di input username e input password verranno composti dall'input che venne inserito dall'utente. In questa composizione di query c'è un rischio che è quello che oltre a inserire le informazioni, che effettivamente vanno inserite come in questo caso, si possa aggiungere altro alla query cambiandole la natura. Vediamo un esempio sempre sulla base di questo esempio qua: se io aggiungo alla query precedente una clausola ulteriore in OR, ad esempio una clausola identica che sarà sempre vera, io posso entrare in un sistema senza conoscere lo username e la password per quel sistema. Quindi bisogna fare attenzione quando si vanno a comporre delle query che poi vengono utilizzate dell'applicativo per dare l'accesso ai dati che effettivamente la query che venga composta contenga solo la query nella forma che era stata programmata originariamente e che non abbia dei componenti che sono estranei alla sua forma originale, che potrebbero causare accessi non desiderati al sistema. Quindi un altro aspetto che sarà da controllare nel controllo degli accessi, oltre al fatto effettivamente di controllare le credenziali o le query che vengono fatte secondo le regole che abbiamo visto finora, è anche che la richiesta dell'utente sia una richiesta legittima dal punto di vista anche della struttura della richiesta che ci si aspetta su quel sistema.