LA SICUREZZA NELLE BASI DI DATI

NICOLA VITACOLONNA, MARIA CONCETTA BROCATO

8.1 Introduzione

Il Laboratorio di Basi di Dati è stato realizzato con l'obiettivo di sensibilizzare gli studenti rispetto a problemi di riservatezza dei dati e di fornire alcune nozioni tecniche fondamentali inerenti alla sicurezza nel contesto specifico dei sistemi di gestione di basi di dati. L'idea di affrontare tale argomento è maturata da una coincidenza di circostanze, alcune soltanto marginalmente legate alla realizzazione concreta dell'attività. Innanzitutto, lo scandalo delle intercettazioni massicce e (non più) segrete effettuate dalla *National Security Agency* americana e da enti governativi di altre nazioni, rese note dall'ormai famoso Edward Snowden. La gravità di questa notizia forse non è stata apprezzata appieno dall'opinione pubblica (soprattutto al di fuori degli USA) — e questo sarebbe già qualcosa su cui bisognerebbe riflettere — ma una parte della comunità IT è (fortunatamente) rimasta profondamente turbata, non tanto perché non si sapesse che il governo americano ha la tendenza a ficcare il naso in tutto ciò che è legato alla sicurezza informatica, quanto per la portata difficilmente immaginabile delle azioni condotte, che hanno "*sovvertito la rete Internet a ogni livello, facendola diventare una vasta, stratificata e robusta piattaforma di sorveglianza*" [99]. L'idea che il problema del *data gate* non sia soltanto etico, politico e sociale, ma anche tecnologico (come si può "ricostruire" Internet in modo che una sorveglianza totale non sia più possibile?) significa che dobbiamo formare una generazione di nuovi scienziati, informatici e ingegneri che abbia sì competenze tecniche, ma anche un'adeguata sensibilità verso tali problematiche.

In secondo luogo, nel periodo in cui mi è stato proposto di svolgere un'attività con le scuole superiori, stavo preparando il materiale del corso di *Complementi di Basi di Dati* che tengo all'Università di Udine, all'interno del quale avevo pensato, sull'onda emozionale degli eventi summenzionati, di dedicare maggior spazio alla parte relativa alla sicurezza. Consultando vari testi, mi sono reso conto che il tema della privatezza dei dati è trattato superficialmente, o è addirittura completamente ignorato[,] in molti libri di testo sulle basi di dati, soprattutto per quanto riguarda gli aspetti più formali. Il Laboratorio mi è sembrato un'occasione appropriata per affrontare tali argomenti, che rivestono una notevole importanza pratica oltre che teorica.

Infine, più o meno nello stesso periodo, mi è capitato tra le mani un articolo intitolato *Hippocratic Databases* [2], che non avevo mai letto prima, in cui si auspica che la futura ricerca nel campo delle basi di dati includa la confidenzialità dei dati come *principio fondante* nella progettazione dei sistemi informatici e nella gestione responsabile delle informazioni. Pur essendo un articolo di dodici anni fa, mi è sembrato estremamente attuale, un ottimo ponte tra le questioni di principio, oggi più che mai messe in discussione, e gli aspetti

tecnologici legati alla realizzazione di sistemi in grado di soddisfare opportuni requisiti di privacy.

8.2 Inquadramento storico

La sicurezza dei dati si è evoluta rapidamente a partire dalla metà degli anni '70 del XX secolo. In quegli anni vi sono stati straordinari avanzamenti nel campo della crittografia (basti pensare all'invenzione della crittografia a chiave pubblica); sono state sviluppate tecniche per verificare in modo formale che un programma non esponga dati confidenziali a soggetti non autorizzati; sono stati studiati in modo sistematico gli attacchi alle basi di dati statistiche. In generale, si è giunti a una migliore comprensione delle limitazioni teoriche e pratiche inerenti alla sicurezza dei dati [93]. In particolare, la ricerca sulle basi di dati statistiche, intensa negli anni '70, ha dimostrato, in modo abbastanza sconfortante, che è impossibile garantire che informazioni confidenziali non trapelino da una base di dati, anche se sono divulgati soltanto dati aggregati [43]. Solo in anni recenti sono stati scoperti nuovi metodi di tipo probabilistico che permettono di dimostrare che una base di dati è "sufficientemente sicura", ossia che i rischi di rivelare informazioni riservate possono essere resi "sufficientemente bassi" [50].

Sempre negli anni '70 è stata definita la nozione di *modello dei dati* [33] ed è stato inventato il *modello relazionale* [34, 32], che è ancora oggi il fondamento della maggior parte dei sistemi di gestione delle basi di dati e che include diverse caratteristiche legate alla sicurezza [35]. Non è sorprendente, perciò, che una parte significativa della ricerca nel campo della sicurezza (concernente ad esempio il controllo degli accessi) abbia trovato un fertile terreno di sviluppo nel contesto dei sistemi relazionali. Gli attuali sistemi di gestione di basi di dati offrono un ampio ausilio alla sicurezza, con funzionalità che comprendono meccanismi di autenticazione, sofisticati controlli sugli accessi ai dati, monitoraggio delle attività degli utenti (*logging* e *auditing*), verifica automatica di vincoli d'integrità, creazione di viste parziali sui dati, sessioni criptate e memorizzazione dei dati in forma cifrata.

Con l'avvento di Internet negli anni '90 e con l'esplosione del commercio elettronico, del *cloud computing* e dei *social network* nel nuovo millennio, la sicurezza dei dati è diventata un tema ancora più cruciale e complesso, sia perché i sistemi hardware e software sono sempre più [sistemi] distribuiti sia perché è aumentata in modo esponenziale la quantità di informazioni che ciascuno di noi condivide attraverso la rete. La ricerca in anni recenti è stata motivata soprattutto dalla crescente tendenza a considerare i sistemi di gestione di basi di dati come *servizi* offerti da organizzazioni esterne. In tale contesto, un argomento di ricerca importante è lo sviluppo di tecniche di processamento delle interrogazioni su dati criptati. Lo sviluppo di applicazioni web cooperative, spesso operanti in contesti più ampi della singola organizzazione o azienda, e di applicazioni mobili, nonché la necessità di

adeguare i sistemi informatici alle nuove leggi sulla privacy promulgate in molti paesi (inclusa l'Italia), hanno fornito un'ulteriore spinta verso lo sviluppo di nuove tecniche di protezione dei dati [12]. Infine, è necessario menzionare la recente voga dei *big data*, che pone nuove questioni di tipo etico, sociale e legale inerenti alla gestione responsabile di enormi quantità di dati sensibili (ad esempio, genomi umani, dati biometrici, sulla localizzazione geografica degli individui, e così via), questioni che costituiscono una delle sfide dell'Informatica nel prossimo futuro.

8.3 Descrizione

Un approccio completo alla sicurezza dei dati richiede di considerare i seguenti tre aspetti [12, 53]:

- l'integrità dei dati;
- la disponibilità dei dati;
- la confidenzialità dei dati.

L'*integrità* si riferisce al fatto che i dati devono essere protetti da modificazioni non consentite (accidentali o meno), vale a dire cambiamenti che comprometterebbero la correttezza delle informazioni memorizzate nella base di dati. Si tratta ovviamente di un requisito fondamentale in molte applicazioni, in cui dati inaccurati possono arrecare gravi danni economici, portare a decisioni errate o aumentare il rischio di frodi. I sistemi di gestione di basi di dati offrono diversi strumenti per la specificazione e la verifica automatica di vincoli d'integrità.

La *disponibilità* riguarda la prevenzione e il recupero a fronte di guasti hardware e software, intenzionali o meno, nonché qualunque forma di attacco a un sistema informatico volta a interromperne il funzionamento. Il problema di garantire un elevato grado di disponibilità delle informazioni è un complesso, interdisciplinare e richiede conoscenze approfondite sulle reti informatiche, i sistemi operativi, le basi di dati e i sistemi distribuiti.

Il terzo aspetto, la *privacy* dei dati, è quello su cui è stata posta l'enfasi nel laboratorio proposto agli studenti, e si riferisce al fatto che l'accesso ai dati deve avvenire sempre e soltanto da parte dei soli utenti autorizzati secondo le modalità previste. La perdita di confidenzialità può avere ripercussioni sociali, etiche e legali, tanto più gravi quanto più sensibili sono i dati trattati (si pensi ad esempio alla diffusione non controllata di dati personali sulla salute).

Diverse tipologie di contromisure possono essere adottate per proteggere i dati da potenziali minacce alla confidenzialità degli stessi:

Controllo dei flussi d'informazione: con quali modalità e attraverso quali canali le informazioni possono passare da un soggetto a un altro?

Controllo degli accessi: quali meccanismi si possono sfruttare per vincolare l'accesso alle risorse da parte degli utenti di un sistema?

Controllo delle inferenze: in che modo è possibile dedurre *indirettamente* informazioni confidenziali [in modo indiretto], conoscendo soltanto dati di carattere statistico su un insieme di individui?

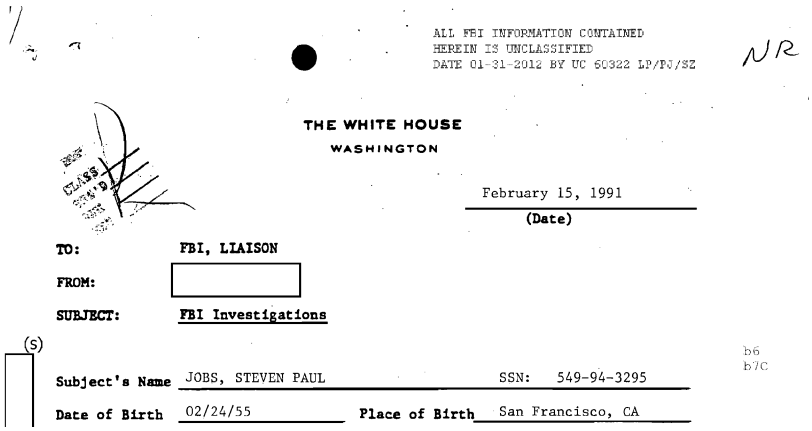
Cifratura dei dati: quali tecniche crittografiche si possono usare per trasmettere e memorizzare informazioni sensibili?

L'uso della crittografia è un tema trasversale rispetto alle basi di dati ed è stato discusso in un laboratorio separato. Perciò, nonostante sia chiaramente un argomento di fondamentale importanza nel contesto della sicurezza dei dati, non è ulteriormente approfondito.

Flussi d'informazione

Una caratteristica interessante delle misure di controllo nelle basi di dati è che, entro una certa misura, possono essere modellate matematicamente in modo relativamente semplice. In termini estremamente generali, possiamo immaginare che alcuni *soggetti* (che possono essere, ad esempio, utenti autorizzati, spie, o anche processi quali applicazioni client legittime oppure software malizioso) operino in qualche modo su un insieme di *oggetti* (ad esempio, documenti cartacei, file del file system o tabelle di una base di dati). Gli oggetti non hanno tutti la stessa "importanza": alcuni possono contenere dati di dominio pubblico, altri [possono contenere] informazioni estremamente sensibili che non devono essere in alcun modo diffuse. A ciascun oggetto, perciò, può essere assegnato un *livello di sicurezza*, che corrisponde in qualche modo al grado di confidenzialità che vogliamo associargli. Intuitivamente, l'informazione non può in generale essere trasferita liberamente da oggetti con un dato livello di sicurezza ad altri con un livello di sicurezza diverso, altrimenti non vi è alcun modo di garantire la confidenzialità delle informazioni. Una *politica di flusso dell'informazione* descrive le regole che stabiliscono in che modo l'informazione possa essere trasferita da un livello di sicurezza ad un altro. Detto $\mathcal{L}$ l'insieme di tutti i possibili livelli di sicurezza, possiamo scrivere $A \rightarrow B$ per indicare che è ammissibile che l'informazione passi da oggetti nel livello di sicurezza A a oggetti nel livello [di sicurezza] B. C'è poi il problema di stabilire quale livello di sicurezza assegnare a un oggetto quando questo è ottenuto combinandone altri (ad esempio, unendo due documenti, uno contenente dati sensibili e uno contenente dati non sensibili). Se indichiamo l'operazione di "combinazione" di due oggetti con $\oplus$, allora possiamo scrivere $A \oplus B = C$ per indicare che il livello di sicurezza di un oggetto ottenuto combinando in qualche modo un oggetto di livello A e uno di livello B dev'essere C.

Con la notazione appena introdotta è possibile specificare in modo preciso varie politiche di flusso dell'informazione. Ad esempio,



ALL FBI INFORMATION CONTAINED
HEREIN IS UNCLASSIFIED
DATE 01-31-2012 BY UC 60322 LP/PJ/SZ

NR

THE WHITE HOUSE
WASHINGTON

February 15, 1991
(Date)

TO: FBI, LIAISON

FROM:

SUBJECT: FBI Investigations

(S)

b6
b7C

Subject's Name JOBS, STEVEN PAUL SSN: 549-94-3295

Date of Birth 02/24/55 Place of Birth San Francisco, CA

Figura 8.1: Un documento dell'FBI con livello di sicurezza *Unclassified*, e dunque di dominio pubblico (Fonte: http://vault.fbi.gov/steve-jobs).

in molte entità governative e militari (e in molti film di spionaggio...) i documenti sono classificati mediante una tra quattro possibili etichette che, in ordine di grado di segretezza decrescente, sono: *Top Secret*, *Secret*, *Confidential*, *Unclassified* (Figura **??**). Ovviamente, non si vuole che un'informazione "top secret" sia scritta all'interno di un documento soltanto "confidenziale", mentre non c'è nessun problema se un'informazione "confidenziale" è scritta in un documento "top secret". In altre parole, il flusso lecito d'informazione è *Unclassified* $\rightarrow$ *Confidential* $\rightarrow$ *Secret* $\rightarrow$ *Top Secret*. Inoltre, se si combinano due documenti con livelli di sicurezza distinti, è naturale che il documento risultante abbia livello di sicurezza pari al più alto dei livelli di sicurezza dei documenti originali. In altre parole, l'operazione $\oplus$ è definita in questo modo:

$$
\begin{aligned}
\textit{Unclassified} \oplus \textit{Unclassified} &= \textit{Unclassified} \\
\textit{Unclassified} \oplus \textit{Confidential} &= \textit{Confidential} \\
\textit{Unclassified} \oplus \textit{Secret} &= \textit{Secret} \\
\textit{Unclassified} \oplus \textit{Top Secret} &= \textit{Top Secret} \\
\textit{Confidential} \oplus \textit{Unclassified} &= \textit{Confidential} \\
\textit{Confidential} \oplus \textit{Confidential} &= \textit{Confidential} \\
\textit{Confidential} \oplus \textit{Secret} &= \textit{Secret} \\
\textit{Confidential} \oplus \textit{Top Secret} &= \textit{Top Secret} \\
&\vdots
\end{aligned}
$$

e così via. Dovrebbe essere chiaro che le operazioni $\rightarrow$ e $\oplus$ non possono essere definite in modo completamente arbitrario, altrimenti si ottengono politiche di flusso dell'informazione prive di senso. In effetti,

una politica di flusso dell'informazione ragionevole deve obbedire a quattro semplici assiomi, detti *assiomi di Denning* [93, 95]:

1. l'insieme $\mathcal{L}$ dei livelli di sicurezza è finito;
2. la relazione $\rightarrow$ è un ordinamento parziale su $\mathcal{C}$;
3. $\mathcal{L}$ ha un estremo inferiore rispetto a $\rightarrow$;
4. $\oplus$ è un operatore totale di estremo superiore.

Cerchiamo di chiarire il significato degli assiomi. La prima proprietà è ovvia: esiste un numero limitato di etichette di sicurezza che è possibile assegnare.

La seconda proprietà richiede che la relazione che determina i flussi d'informazione sia un *ordinamento parziale*, ossia una relazione *riflessiva* (ogni elemento è in relazione con sé stesso), *transitiva* (se x è in relazione con y e y è in relazione con z allora x è in relazione con z) e *antisimmetrica* (se x è in relazione con y e y è in relazione con x allora x e y sono uguali). Ciascuna di queste proprietà ha un'interpretazione intuitiva in termini di requisiti di sicurezza:

- riflessività: per ogni livello di sicurezza A si ha $A \rightarrow A$, ossia l'informazione può fluire liberamente tra oggetti allo stesso livello di sicurezza;
- transitività: se $A \rightarrow B$ e $B \rightarrow C$ allora $A \rightarrow C$, ossia se l'informazione può fluire indirettamente dal livello A al livello C attraverso B, allora la stessa informazione può passare direttamente da A a C;
- antisimmetria: se $A \rightarrow B$ e $B \rightarrow A$ allora $A = B$, ossia è inutile avere livelli di sicurezza distinti se l'informazione può passare liberamente dall'uno all'altro.

Anche il terzo assioma esprime una proprietà abbastanza intuitiva: esiste un livello di sicurezza minimo[,] tale che l'informazione può sempre passare da quel livello a qualunque altro [livello] (ad esempio[,] un livello di sicurezza per informazioni di pubblico dominio).

L'ultimo assioma è meno banale. Che cosa significa che $\oplus$ è un "operatore totale"? Vuol dire che è definito per ogni possibile coppia di livelli di sicurezza, e dunque è sempre possibile assegnare un livello di sicurezza all'oggetto risultante dalla combinazione di altri due oggetti. Il fatto che $\oplus$ debba essere l'operatore di "estremo superiore" cattura in modo preciso l'intuizione dell'esempio precedente, cioè essenzialmente che la combinazione di due documenti non può avere un livello di sicurezza più basso di uno dei due documenti originali, e non ha senso che abbia un livello di sicurezza strettamente più alto. Più formalmente, si deve avere che

- $A \rightarrow A \oplus B$ e $B \rightarrow A \oplus B$;
- se $A \rightarrow C$ e $B \rightarrow C$ allora $A \oplus B \rightarrow C$;

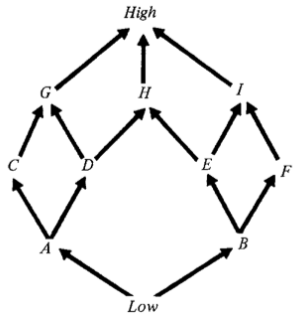


Figura 8.2: Un esempio di reticolo [(Fonte:][93][).]

- $\oplus$ è commutativo (cioè[,] $A \oplus B = B \oplus A$) e associativo (cioè[,] $(A \oplus B) \oplus C = A \oplus (B \oplus C)$).

Si può dimostrare che dalle precedenti proprietà discende che le politiche di flusso dell'informazione sono strutture matematiche chiamate *reticoli* (Figura **??**). Questo è un fatto interessante, perché i reticoli sono ben noti ai matematici (e agli informatici — almeno ad alcuni). La formalizzazione matematica delle politiche di flusso dell'informazione apre perciò la porta allo studio rigoroso delle loro proprietà e permette di capirne i vantaggi e i limiti. Ad esempio, la specificazione formale dei requisiti di sicurezza richiesti da un sistema permette, almeno in linea di principio, di validare i meccanismi di controllo degli accessi ai dati e individuare eventuali vulnerabilità. Un limite è dato dal fatto che, sebbene sia possibile dimostrare il rispetto di una politica di sicurezza rispetto ai flussi d'informazione noti, in pratica bisogna tenere in considerazione anche potenziali canali di scambio dell'informazione impliciti o *nascosti*. Ad esempio, un soggetto ostile potrebbe inferire il testo di un documento senza aver la possibilità di leggerlo direttamente, ma soltanto registrando il suono del ticchettio della tastiera dell'utente che lo sta scrivendo. Un altro esempio di flusso implicito d'informazione è proposto nella sezione successiva.

Si può infine dimostrare che il problema di controllare che tutti e soli i flussi d'informazione autorizzati siano possibili è molto più complesso di quanto le presenti note possano lasciar intendere (in generale è indecidibile!) [93]. La matematica richiesta in questo ambito può diventare piuttosto sofisticata (un ruolo importante è giocato dalla *teoria dell'informazione* di Claude Shannon) e ha numerose applicazioni, ad esempio in crittografia.

Controllo degli accessi

La protezione dei dati rispetto ad accessi non autorizzati è possibile grazie alle funzionalità messe a disposizione dai sistemi di gestione di basi di dati [35, 12]. In particolare, tali sistemi forniscono opportuni *meccanismi di controllo degli accessi*, il cui obiettivo è implementare

una data politica di flusso dell'informazione che garantisca l'accesso alle risorse disponibili solo ai soggetti autorizzati e secondo le regole stabilite [dalla politica]. Tali meccanismi si basano sull'idea di assegnare ai soggetti opportune *autorizzazioni*[,] o *privilegi* (ad esempio di lettura, scrittura, creazione, cancellazione[,] e così via) per operare sugli oggetti della base di dati.

Uno dei più semplici e più diffusi meccanismi di sicurezza prende il nome di *accesso discrezionale* (*discretionary access control* o *DAC*, in inglese). In tale modello, ogni oggetto della base di dati è assegnato a un soggetto *proprietario* (che, tipicamente, è l'utente che ha creato l'oggetto). Il proprietario ha tutti i privilegi sugli oggetti che possiede e può concedere alcuni di tali privilegi ad altri soggetti. La concessione di un privilegio su un oggetto è pertanto a discrezione del proprietario, da cui il nome dato al modello.

Il modello discrezionale non è usato solo nelle basi di dati, ma anche per la gestione dei permessi sui file nei sistemi operativi. Il difetto principale di tale modello è che non preclude la possibilità di flussi d'informazione indesiderati (intenzionali o accidentali). Per illustrare tale affermazione, useremo per semplicità un esempio riferito all'accesso a un file su disco [95].

Supponiamo che Anna, Biagio e Carla siano tre colleghi di lavoro che hanno accesso a uno stesso computer. Supponiamo inoltre che Biagio abbia scritto un documento per Anna in cui esprime un giudizio fortemente negativo sul modo di lavorare di Carla. Biagio vuol far leggere il documento ad Anna, ma non a Carla. Inizialmente, dunque, la situazione potrebbe essere la seguente:

	Documento
Anna	-
Biagio	proprietario
Carla	-

Poiché Biagio ha creato il documento, ne è il proprietario. Gli altri utenti non hanno alcun privilegio su di esso e in particolare non possono leggerlo. Dopo aver creato il documento, Biagio decide di concedere il privilegio di lettura ad Anna:

	Documento
Anna	lettura
Biagio	proprietario
Carla	-

Ora Anna può leggere la lamentela di Biagio, ma Carla ancora no. Tuttavia, nell'account di Anna è presente un *malware* (installato da Carla?), che crea automaticamente una copia di ogni documento leggibile da Anna e ne modifica i privilegi. Poiché la copia del documento è creata dal malware che agisce con i privilegi di Anna, Anna ne diventa proprietaria:

	Documento	Copia del documento
Anna	lettura	proprietario
Biagio	proprietario	-
Carla	-	-

Il malware può quindi concedere a Carla il privilegio di lettura sulla *copia* del documento:

	Documento	Copia del documento
Anna	lettura	proprietario
Biagio	proprietario	-
Carla	-	lettura

cosicché Carla è ora in grado di sapere ciò che Biagio pensa di lei.

Per superare le limitazioni dell'accesso discrezionale, è necessario definire politiche d'accesso a livello di sistema, che abbiano priorità rispetto ai privilegi discrezionali e che gli utenti non siano in grado di modificare. Una soluzione è data da un modello basato sull'*accesso vincolato* (*mandatory access control* o *MAC*, in inglese), inizialmente proposto per la sicurezza di sistemi militari. In tale modello, a ciascun soggetto è assegnato un *livello d'autorizzazione*, in modo analogo a quanto abbiamo visto in precedenza per gli oggetti con i livelli di sicurezza. Intuitivamente, soggetti con un elevato livello d'autorizzazione sono fidati e si assume che non facciano filtrare informazioni riservate (in altre parole, tale modello non avrebbe potuto impedire a Edward Snowden di confermarci che viviamo in un presente orwelliano). Per acquisire un certo livello d'autorizzazione devono essere poste in atto opportune procedure di verifica di affidabilità degli utenti. Per contro, i sistemi software potrebbero contenere malware, perciò, in generale, a essi dev'essere assegnato un basso livello d'autorizzazione.

L'accesso vincolato si basa su due regole, dette *regole di Bell-LaPadula* [9, 8]:[1]. Se S è un soggetto e O è un oggetto allora deve valere quanto segue:

1. S può leggere O solo se il livello d'autorizzazione di S è almeno pari al livello di sicurezza di O (no read up);

2. S può scrivere O solo se il livello di sicurezza di O è almeno pari al livello di autorizzazione di S (no write down).

In termini di flussi d'informazione:

1. la prima regola implica un flusso d'informazione da un oggetto O a un soggetto S con sufficienti privilegi;

2. la seconda regola implica un flusso d'informazione da un soggetto S a un oggetto O sufficientemente sicuro.

[1] Il modello è notevolmente semplificato rispetto alla proposta originale.

Ad esempio, un soggetto con livello d'autorizzazione *Secret* può leggere oggetti classificati come *Secret*, *Confidential* o *Unclassified*, ma non oggetti *Top Secret*, [e] dunque non può accedere in alcun modo alle informazioni più riservate; un soggetto con accesso *Confidential* può scrivere oggetti *Confidential*, *Secret* e *Top Secret* (anche se non può leggere da questi ultimi due!), ma non può scrivere oggetti *Unclassified* e dunque non può, nemmeno involontariamente, rivelare informazioni sensibili.

Vediamo come l'accesso vincolato permetta a Biagio di comunicare in maniera sicura con Anna. Ciò può accadere se Biagio e Anna hanno un livello d'autorizzazione più alto rispetto a Carla. Ad esempio, a Biagio e Anna potrebbe essere stato assegnato il livello d'autorizzazione *Secret*, mentre Carla potrebbe avere soltanto il livello *Confidential*. Quando Biagio crea il documento per Anna, può etichettarlo come *Secret* (ciò è consistente con la seconda regola di Bell-LaPadula):

	Documento (*Secret*)
Anna (*Secret*)	
Biagio (*Secret*)	proprietario
Carla (*Confidential*)	

Come prima, Biagio assegna il privilegio di lettura ad Anna, e come prima il malware installato nell'account di Anna crea una copia del documento e dà il privilegio di lettura a Carla:

	Documento (*Secret*)	Copia del documento (*Secret*)
Anna (*Secret*)	lettura	proprietario
Biagio (*Secret*)	proprietario	
Carla (*Confidential*)		lettura

Tuttavia, in base alla seconda regola di Bell-LaPadula, il malware può creare solo documenti con livello di sicurezza *Secret* (o *Top Secret*). In base alla prima regola, Carla non può leggere tali documenti, indipendentemente dai permessi assegnati al documento. Di conseguenza, la privacy della comunicazione tra Biagio e Anna è salvaguardata.

Controllo degli accessi in SQL

Lo standard SQL prevede due istruzioni [rispettivamente] per l'assegnamento e la revoca di privilegi: *rispettivamente* `grant` e `revoke`. I principali privilegi sono `connect` (connessione a una base di dati), `select` (lettura di record), `insert` (inserimento di nuovi record), `update` (aggiornamento di record esistenti), `delete` (cancellazione di record) ed `execute` (possibilità di eseguire funzioni definite dall'utente). È inoltre possibile specificare se l'utente che riceve i privilegi può propagarli ad altri utenti oppure no. Ad esempio, l'istruzione:

```
grant select on T to Giorgio;
```

assegna a Giorgio il privilegio di lettura sull'oggetto T; Giorgio, tuttavia, non può concedere tale privilegio ad altri utenti. Invece l'istruzione:

```
grant select on T to Giorgio with grant option;
```

consente a Giorgio di assegnare a terzi il privilegio ricevuto.

L'esempio basato su DAC della sezione precedente può essere riformulato in termini di operazioni sulle tabelle di una base di dati. Assumiamo che Anna, Biagio e Carla siano tre utenti che hanno accesso alla stessa base di dati. L'esempio può essere allora codificato in SQL come segue:

1. Biagio crea una tabella e v'inserisce il documento per Anna. Biagio è il proprietario della tabella e ha automaticamente tutti i privilegi su di essa, mentre gli altri utenti non hanno alcun privilegio d'accesso [sulla tabella]:

   ```
   create table Documenti(
     id int primary key,
     testo varchar(1000)
   );

   insert into Documenti(id, testo) values (1, '...');
   ```

2. Biagio assegna ad Anna il privilegio di lettura sulla tabella:

   ```
   grant select on Documenti to Anna;
   ```

   Ora Anna può collegarsi alla base di dati e leggere il contenuto della tabella:

   ```
   table Documenti;

    id | testo
   ----+-------
     1 | ...
   ```

3. A questo punto, un malware in grado di collegarsi alla base di dati con i privilegi di Anna crea una copia della tabella:

   ```
   create table CopiaDocumenti as (
     select * from Documenti
   );
   ```

4. Poiché la tabella `CopiaDocumenti` è di proprietà di Anna, il malware (che opera con i privilegi di Anna) può modificarne a piacimento i privilegi:

   ```
   grant select on CopiaDocumenti to Carla;
   ```

   Si noti che il malware non può propagare direttamente i privilegi della tabella `Documenti`, perché ad Anna non sono stati concessi con la clausola `with grant option`.

5. Ora Carla può collegarsi alla base di dati e leggere il contenuto della nuova tabella:

```
table CopiaDocumenti;


 id | testo
----+-------
  1 | ...
```

Mentre il DAC è solitamente presente nella maggior parte dei moderni sistemi basati su SQL, il modello MAC di solito non è direttamente supportato. Tuttavia, alcuni sistemi di gestione dei dati (in particolare, PostgreSQL, [il sistema] usato nel laboratorio) implementano un meccanismo di controllo degli accessi molto flessibile, detto *controllo degli accessi basato sui ruoli* (*role-based access control* o *RBAC* in inglese).

Il modello RBAC si basa sul concetto di *ruolo*, che può essere pensato sia come un singolo utente della base di dati sia come un gruppo di utenti. Ad esempio, `antonio` può essere il nome di un ruolo che corrisponde a una persona di nome Antonio, mentre `studente` può essere il nome di un ruolo che include tutti gli utenti che sono studenti. Il sistema non distingue tra ruoli che corrispondono a utenti oppure a gruppi: la distinzione è fatta dall'amministratore della base di dati tenendo conto del significato del ruolo stesso.

Ciascun ruolo può essere proprietario di uno o più oggetti della base di dati e può assegnare ad altri ruoli i privilegi d'accesso su tali oggetti. Esiste poi una nozione di *appartenenza* di un ruolo ad un altro: ad esempio, un ruolo `tecnico` può appartenere a un ruolo `impiegato`: ciò significa che i privilegi assegnati a un impiegato sono ereditati da ogni tecnico. In altre, parole, è possibile costruire *gerarchie di ruoli*. Si può dimostrare che RBAC è sufficientemente flessibile da poter essere usato per implementare sia il modello DAC sia il MAC [85].

Discutiamo come simulare MAC con RBAC mediante un esempio. Si consideri la politica di flusso dell'informazione descritta dal reticolo in Figura **??**. In accordo al modello di Bell-LaPadula, i soggetti con livello di sicurezza più elevato hanno maggiori autorizzazioni in lettura (ad esempio[,] al livello *Segreto* un soggetto ha accesso in lettura a qualunque oggetto), ma hanno anche maggiori vincoli in scrittura (al livello *Segreto* è possibile scrivere soltanto oggetti classificati come *Segreti*). Il duplice carattere del flusso d'informazione può essere catturato da due gerarchie di ruoli, una per le operazioni di lettura e una per quelle di scrittura, come mostrato in Figura **??** (in cui il simbolo $\in$ denota l'appartenza di un ruolo ad un altro, dunque specifica che il ruolo membro eredita i privilegi del ruolo cui appartiene e può eventualmente averne di ulteriori). A ciascun livello L della Figura **??** si fa corrispondere una coppia di ruoli L_r e L_w. Intuitivamente, al ruolo L_r sono assegnati i privilegi di lettura degli oggetti con livello di sicurezza L; analogamente a L_w sono assegnati i privilegi di modificazione degli oggetti con livello di sicurezza L. La struttura delle

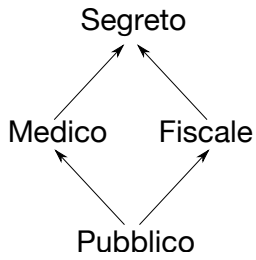


Figura 8.3: Un esempio di politica di flusso dell'informazione.

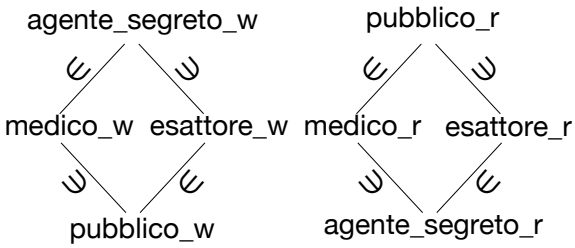


Figura 8.4: Gerarchie di ruoli corrispondenti alla politica di Figura ??.

gerarchie garantisce il rispetto delle regole di Bell-LaPadula. Infine, ciascun utente della base di dati è assegnato a uno e un solo ruolo di tipo L_r e a uno e uno solo di tipo L_w, forzando in tal modo il rispetto della politica definita.

Controllo delle inferenze

Una *base di dati statistica* è una base di dati in cui sono consentite soltanto interrogazioni che producono dati aggregati (ad esempio, somme o medie di insiemi di valori). Le interrogazioni che consentono di derivare informazione di tipo individuale sono bloccate dal sistema. Un'interrogazione che non è permessa in una base di dati statistica è, ad esempio, la seguente:

```
select * from Persona where cod_fiscale ='X';
```

La precedente interrogazione consente di ottenere tutte le informazioni inerenti a uno specifico individuo (quello con codice fiscale X). Al contrario, un'interrogazione del tipo

```
select avg(stipendio) from Persona;
```

potrebbe essere ammissibile, perché produce un dato (lo stipendio medio) che, in generale, non è riferito a un individuo in particolare.

Il problema principale delle basi di dati statistiche è che quasi sempre è possibile dedurre le risposte a interrogazioni non ammissibili in modo indiretto formulando soltanto interrogazioni ammissibili [43]. Illustreremo il problema mediante un esempio ispirato a [39].

Consideriamo una base di dati che contiene una tabella con il seguente schema:

Impiegato(nome, sesso, professione, stipendio)

Supponiamo di sapere che uno degli impiegati è un programmatore di nome Alfio. Vorremmo conoscere lo stipendio di Alfio, ma il sistema non accetta alcuna interrogazione del tipo:

```
select stipendio from Persona where nome = 'Alfio';
```

perché è permesso calcolare soltanto risultati aggregati. Senza darci per vinti, proviamo a formulare la seguente interrogazione:

```
select count(*) from Impiegato
  where sesso = 'M' and professione = 'programmatore';
```

Supponiamo che il sistema dia come risultato 1. Abbiamo scoperto che c'è un solo programmatore maschio: deve trattarsi di Alfio! Ecco come possiamo scoprire il suo stipendio:

```
select sum(stipendio) from Impiegato
  where sesso = 'M' and professione = 'programmatore';
```

Dato il risultato dell'interrogazione precedente, sappiamo che la somma calcolata da questa interrogazione è la somma di un singolo stipendio.

L'esempio precedente può sembrare estremamente *ad hoc* e si potrebbe argomentare che siamo stati soltanto fortunati, perché abbiamo trovato una condizione che permette di isolare un singolo record della tabella. Una semplice contromisura consiste nel vietare di rispondere a interrogazioni che coinvolgono pochi record (e, simmetricamente, a interrogazioni che li coinvolgono quasi tutti). Si può dimostrare, tuttavia, che tali restrizioni sono completamente inefficaci: è possibile definire condizioni di selezione che consentono di ottenere una risposta (indiretta) praticamente a qualunque interrogazione inammissibile. Tali condizioni sono dette *tracker generali* [43, 39].

L'idea è abbastanza semplice. Intuitivamente, un tracker generale è una condizione che è vera per "circa la metà" dei record di una tabella. Se una condizione A è inammissibile, ad esempio perché è vera per un numero troppo piccolo di record, allora nelle interrogazioni si può usare A or T invece di A, dove T è un tracker generale. La condizione A or T risulta essere sempre ammissibile, perché sarà vera per "circa la metà più qualcosa" dei record, quindi né per un numero troppo piccolo né per un numero troppo grande. Inoltre, se T è un tracker generale, anche not T lo è, perciò anche la condizione A or not T è sempre ammissibile.

Più precisamente, se una tabella ha N record e le interrogazioni ammissibili sono solo quelle che aggregano insiemi di valori di cardinalità compresa tra k e $N - k$ per qualche valore k fissato, allora ogni condizione soddisfatta da almeno $2k$ e da non più di $N - 2k$ record ("circa la metà" significa questo) è un tracker generale (questo fatto richiede ovviamente una dimostrazione, che omettiamo).

Per verificare se una condizione è un tracker generale, basta eseguire due interrogazioni del tipo:

```
select count(*) from Persona where <condizione>;
select count(*) from Persona where not <condizione>;
```

Se entrambe le interrogazioni sono accettate dal sistema e producono un risultato compreso tra k e $N - k$, allora la condizione è un tracker generale. Non solo, ma la somma dei risultati delle due interrogazioni permette di conoscere quanti record ci sono nella tabella.

Se si ha la possibilità di formulare tali interrogazioni in un dato sistema, non è particolarmente difficile trovare un tracker generale, anche procedendo per tentativi. Ad esempio, dopo che sono state spiegate queste tecniche agli studenti, diversi di loro sono riusciti a risolvere il seguente problema: dato lo schema

Finanziamento(nome, sesso, professione, contributo)

che memorizza informazioni relative a finanziamenti segreti a partiti politici, sapendo che Daria è una giornalista corrotta, quanti soldi ha dato ai partiti? (L'istanza usata è riportata in appendice.)

8.4 La voce della scuola

Lo scopo principale del Piano Lauree Scientifiche è quello di orientare gli studenti [alle future scelte personali] tramite azioni comuni tra Scuola Secondaria ed Università. In questo contesto di prospettiva verso il futuro e di scoperta la Scuola Secondaria, oltre a realizzare l'obiettivo primario, può trovare ampi spazi per svolgere azioni didattiche innovative. L'attività di tipo laboratoriale permette infatti al docente [referente scolastico] ed agli studenti una *flessibilità di ruoli*:

- il docente assume un ruolo di guida del percorso di costruzione del sapere dello studente ma nel contempo, *scendendo dalla cattedra*, può dedicarsi alla ricerca, alla progettazione e sperimentazione di nuove modalità didattiche che prevedono la collaborazione ed supporto concreto degli studenti stessi;

- gli studenti, *possono non essere fruitori passivi*: sono liberi di dedicare maggiormente l'attenzione ad alcuni aspetti del percorso, ricercano soluzioni innovative e possono proporre in prima persona attività al docente, realizzando un percorso personalizzato.

In quest'ottica di sperimentazione e con una sinergia tra il docente di classe ed il docente dell'Università si è svolta l'attività *"La sicurezza delle basi di dati: Spie, cavalli di Troia e basi di dati ippocratiche"*, un percorso di esplorazione che ha permesso di svolgere anche alcune attività collaborative tra studenti: in presenza (in orario extrascolastico), attraverso l'utilizzo della tecnologia (*Moodle*, *canale Youtube*) e di ambienti cooperativi (*wiki*, *forum*).

Il seminario di introduzione alla tematica della sicurezza delle basi di dati tenuto dal dott. N. Vitacolonna, è stato esteso anche ad altre classi dell'Istituto interessate alla tematica. In seguito alcuni studenti hanno partecipato a due laboratori presso l'Università. L'attività di documentazione e pubblicazione di un sito web di descrizione del percorso è stata svolta e coordinata all'interno della scuola.

La conferenza, iniziata con la visione di un breve e divertente cartoon ispirato alla satira politica del film "Dottor Stranamore", è stata affiancata da una attività di *live twitting* con *hashtag* identificati in maniera collaborativa dagli studenti della classe (*#MalignaniUd #PLSDB*) e comunicati alla platea [all'inizio] per permettere a tutti gli ascoltatori di esprimere le loro partecipazione. Dopo il videoclip, che ha immediatamente catturato l'attenzione dei presenti, il relatore ha trattato alcuni importanti aspetti della storia contemporanea legati alla privacy: il recente blocco di alcuni social in Turchia mediante dirottamento dei DNS, gli incontri segreti tra NeXT e l'NSA del '95, i bug di sicurezza inseriti intenzionalmente nella rete Internet e nei dispositivi *mobile* fino ad arrivare, a ritroso nel tempo, a citare il cardinale Richelieu con la celebre frase: "Datemi sei righe scritte per mano dal più onesto degli uomini e ci troverò un motivo per farlo impiccare".

Nella parte dell'incontro più tecnica e legata all'informatica, il relatore ha illustrato alcune delle più comuni tecniche di "*SQL Injection*", ha presentato alcune metodologie di controllo degli accessi ed alcuni esempi di tecniche di valutazione della sicurezza di basi di dati reali. Molto importante, per le implicazioni etiche, la parte finale dedicata alle "Basi di dati ippocratiche" introdotte da Agrawal et al. [2] a partire dal 2002 ed ispirate ad un passo del giuramento di Ippocrate. Attualizzando e trasferendo il giuramento dal campo medico ai moderni database, Agrawal et al. hanno declinato i principi fondanti sui quali si dovrebbe basare un sistema di gestione di una base dati: mettere in primo piano la gestione responsabile dei dati e garantire la privacy dell'individuo.

Per svolgere le successive attività laboratoriali, il gruppo classe è stato suddiviso in team per aree di interesse:

- partecipazione ai due laboratori presso l'Università con attività su database PostgreSQL e progettazione di una base dati con ruoli e privilegi utente specifici.
  Finalità: apprendere il funzionamento di un modello di controllo degli accessi (*Role-Based Access Control*) e implementare una politica di flusso dell'informazione usando tale modello;

- progettazione, scrittura e pubblicazione di un sito HTML/CSS per descrivere le attività all'interno del PLS utilizzando l'ambiente PhpMyAdmin, una base dati normalizzata per la raccolta dei Tweet e l'interrogazione dinamica alla stessa.
  Finalità: approfondire tematiche trattate durante l'anno scolastico ed applicarle ad un caso reale;

- creazione di documentazione progettuale per la relazione finale ed editing di una presentazione delle attività utilizzando *modalità collaborative* (software Prezi, Google Drive).
  Finalità: svolgere attività collaborative di supporto al team di progetto;

- registrazione di uno streaming video del seminario in presenza e contestuale pubblicazione su un canale YouTube.
  Finalità: sperimentazione di attività multimediali di supporto al team di progetto.

I tweet raccolti dagli allievi durante le attività, i contenuti testuali e multimediali, i file ed i concetti ritenuti "chiave" dagli studenti stessi sono stati raccolti e condivisi nel [corso] Moodle della classe e su alcune cartelle di Google Drive. Alcuni dei materiali sono stati pubblicati nel sito progettato nel corso delle attività, che potrà essere fruito nei prossimi anni dagli studenti della scuola. Il modulo PLS è stato inteso, oltre che in ottica orientante, anche come laboratorio collaborativo, per permettere la condivisione del sapere a tutta la classe ed elevare il livello complessivo delle conoscenze.

Tutta l'attività ha favorito il coinvolgimento degli studenti che da "*passivi ascoltatori*" sono diventati *parte attiva*. Durante le attività interne alla scuola "*gli allievi sono stati il motore*", in questo contesto il compito dell'insegnante è stato di guida per "*far emergere le potenzialità e gli interessi*" degli allievi, in stretta collaborazione con il docente referente dell'Università.

Si riportano di seguito i link ad alcuni dei materiali pubblicati e prodotti dalla classe durante il laboratorio:

- Account Twitter Malignani Unofficial creato per il *livetwitting* delle conferenze PLS: https://twitter.com/MalignaniUD
  Hashtag dedicati: *#MalignaniUd #PLSDB*

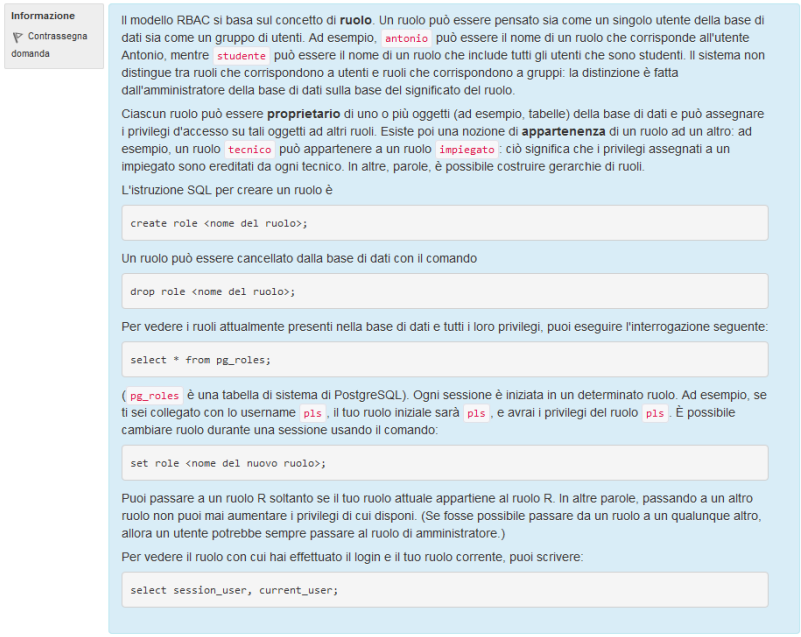
- Sito web progettato e realizzato con utilizzo di linguaggio HTML, fogli di stile CSS, database MySQL:
  http://pls2014.winfuture.it/

- Diretta streaming del seminario organizzata dagli studenti e pubblicata sul canale Youtube visionabile al link:
  http://goo.gl/MUZRGs

- Presentazione dell'attività collaborativa alla manifestazione "GO On FVG" (Liceo Stellini, 5 maggio 2014) a cura della docente di classe, con testimonianza degli allievi S. Cragnolini, I. Manfredi, R. Nobile, A. Roccaforte:
  https://goo.gl/9qtpcW (testo)
  https://goo.gl/s1XcOZ (video)

- News pubblicata sul sito scolastico:
  http://goo.gl/nQ1y4J

Esempi di appunti e documenti condivisi dagli studenti:



Informazione

Contrassegna domanda

Il modello RBAC si basa sul concetto di **ruolo**. Un ruolo può essere pensato sia come un singolo utente della base di dati sia come un gruppo di utenti. Ad esempio, `antonio` può essere il nome di un ruolo che corrisponde all'utente Antonio, mentre `studente` può essere il nome di un ruolo che include tutti gli utenti che sono studenti. Il sistema non distingue tra ruoli che corrispondono a utenti e ruoli che corrispondono a gruppi: la distinzione è fatta dall'amministratore della base di dati sulla base del significato del ruolo.

Ciascun ruolo può essere **proprietario** di uno o più oggetti (ad esempio, tabelle) della base di dati e può assegnare i privilegi d'accesso su tali oggetti ad altri ruoli. Esiste poi una nozione di **appartenenza** di un ruolo ad un altro: ad esempio, un ruolo `tecnico` può appartenere a un ruolo `impiegato`: ciò significa che i privilegi assegnati a un impiegato sono ereditati da ogni tecnico. In altre, parole, è possibile costruire gerarchie di ruoli.

L'istruzione SQL per creare un ruolo è

```
create role <nome del ruolo>;
```

Un ruolo può essere cancellato dalla base di dati con il comando

```
drop role <nome del ruolo>;
```

Per vedere i ruoli attualmente presenti nella base di dati e tutti i loro privilegi, puoi eseguire l'interrogazione seguente:

```
select * from pg_roles;
```

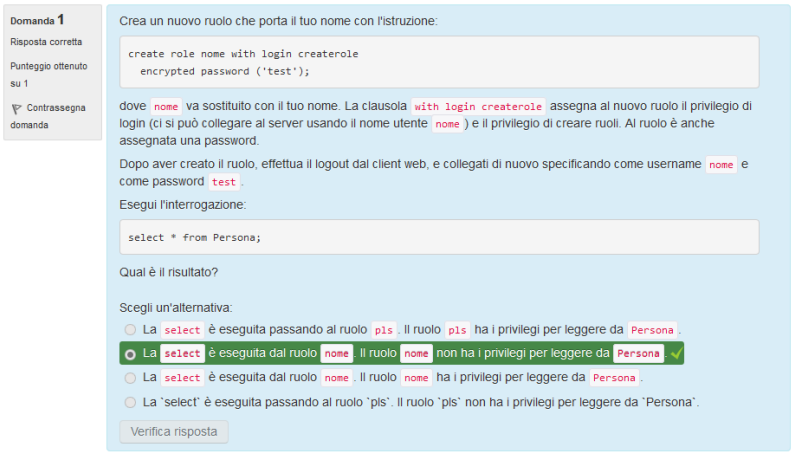
(`pg_roles` è una tabella di sistema di PostgreSQL). Ogni sessione è iniziata in un determinato ruolo. Ad esempio, se ti sei collegato con lo username `pls`, il tuo ruolo iniziale sarà `pls`, e avrai i privilegi del ruolo `pls`. È possibile cambiare ruolo durante una sessione usando il comando:

```
set role <nome del nuovo ruolo>;
```

Puoi passare a un ruolo R soltanto se il tuo ruolo attuale appartiene al ruolo R. In altre parole, passando a un altro ruolo non puoi mai aumentare i privilegi di cui disponi. (Se fosse possibile passare da un ruolo a un qualunque altro, allora un utente potrebbe sempre passare al ruolo di amministratore.)

Per vedere il ruolo con cui hai effettuato il login e il tuo ruolo corrente, puoi scrivere:

```
select session_user, current_user;
```



Domanda 1

Risposta corretta

Punteggio ottenuto su 1

Contrassegna domanda

Crea un nuovo ruolo che porta il tuo nome con l'istruzione:

```
create role nome with login createrole
  encrypted password ('test');
```

dove `nome` va sostituito con il tuo nome. La clausola `with login createrole` assegna al nuovo ruolo il privilegio di login (ci si può collegare al server usando il nome utente `nome`) e il privilegio di creare ruoli. Al ruolo è anche assegnata una password.

Dopo aver creato il ruolo, effettua il logout dal client web, e collegati di nuovo specificando come username `nome` e come password `test`.

Esegui l'interrogazione:

```
select * from Persona;
```

Qual è il risultato?

Scegli un'alternativa:

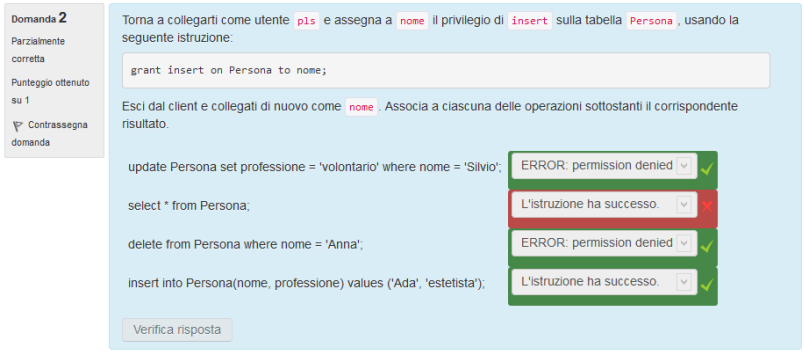
- La `select` è eseguita passando al ruolo `pls`. Il ruolo `pls` ha i privilegi per leggere da `Persona`.
- La `select` è eseguita dal ruolo `nome`. Il ruolo `nome` non ha i privilegi per leggere da `Persona`. ✓
- La `select` è eseguita dal ruolo `nome`. Il ruolo `nome` ha i privilegi per leggere da `Persona`.
- La `select` è eseguita passando al ruolo `pls`. Il ruolo `pls` non ha i privilegi per leggere da `Persona`.

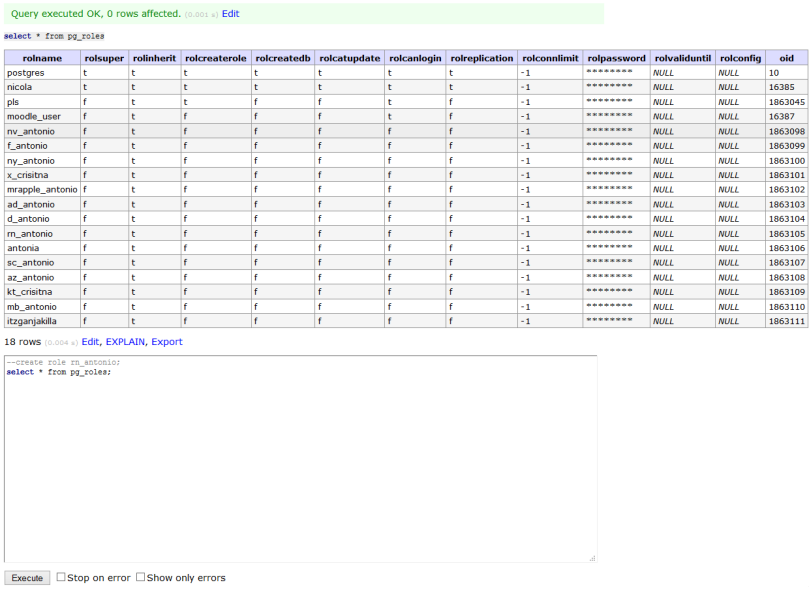
Verifica risposta

- https://goo.gl/1NMQHX
- https://goo.gl/3st8zx
- https://goo.gl/iPtySz

8.5 Conclusioni

Che cos'è la privacy? Una definizione interessante è la seguente: *"La privacy è il diritto di un individuo a determinare da sé quando, come ed entro che limiti l'informazione che lo riguarda è comunicata ad*





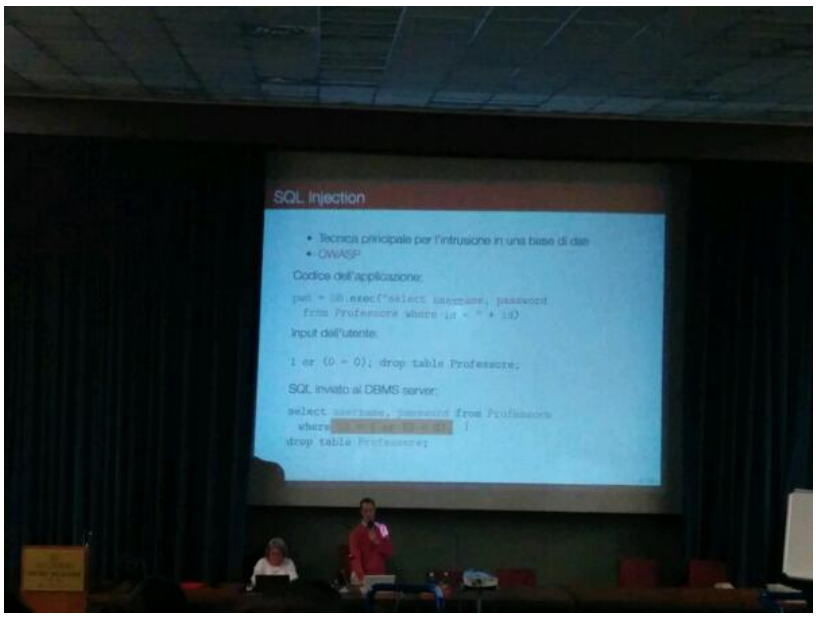
rolname	rolsuper	rolinherit	rolcreaterole	rolcreatedb	rolcatupdate	rolcanlogin	rolreplication	rolconnlimit	rolpassword	rolvaliduntil	rolconfig	oid
postgres	t	t	t	t	t	t	t	-1	********	NULL	NULL	10
nicola	t	t	t	t	t	t	t	-1	********	NULL	NULL	16385
pls	f	t	t	f	f	t	f	-1	********	NULL	NULL	1863045
moodle_user	f	t	f	f	f	t	f	-1	********	NULL	NULL	16387
nv_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863098
f_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863099
ny_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863100
x_crisitna	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863101
mrapple_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863102
ad_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863103
d_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863104
rn_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863105
antonia	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863106
sc_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863107
az_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863108
kt_crisitna	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863109
mb_antonio	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863110
itzganjakilla	f	t	f	f	f	f	f	-1	********	NULL	NULL	1863111

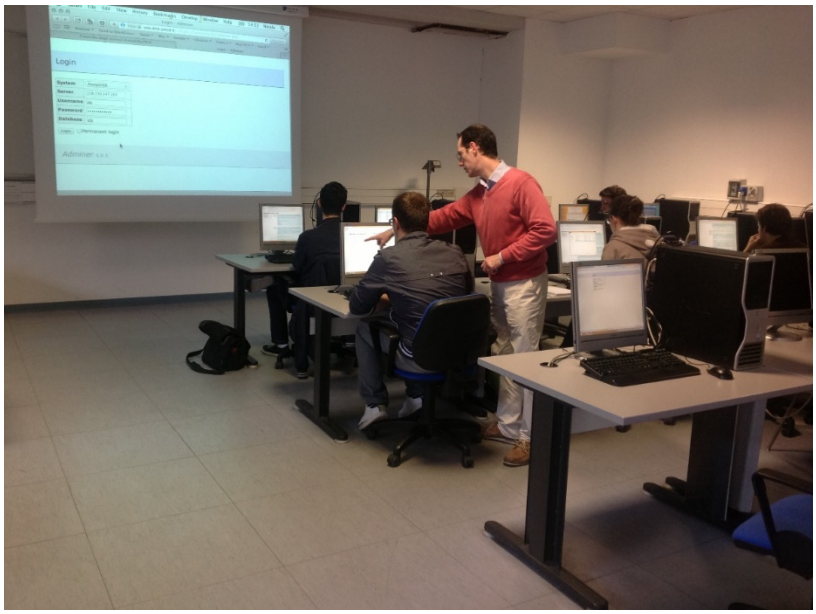
altri" [113]. La privacy *non* è il bisogno di nascondere qualcosa di sbagliato o illecito, come alcuni assumono, tacitamente o in modo esplicito ("Se non fai niente di male, che cos'hai da nascondere?"). Si tratta di un *"diritto umano intrinseco, e un requisito per mantenere la condizione umana con dignità e rispetto* [98]. La piena realizzazione di tale diritto richiede scelte politiche oculate e il supporto della tecnologia. Poiché i sistemi informativi che memorizzano i nostri dati sono basati su sistemi di gestione di basi di dati, è necessario che la progettazione di tali sistemi tenga conto, fin dall'inizio, dei requisiti di privacy [2]:

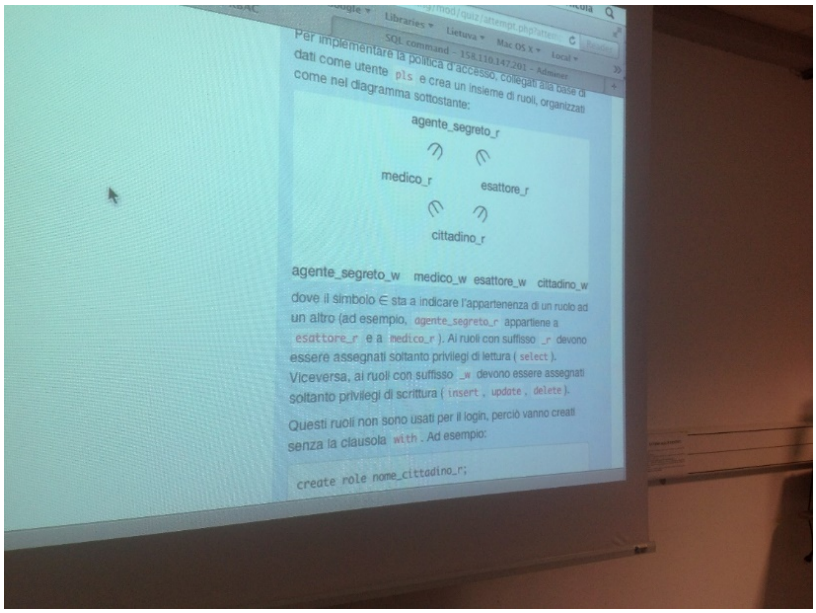
- **Scopi:** per ogni informazione memorizzata nella base di dati devono essere specificate le ragioni per cui [tale informazione] è presente;

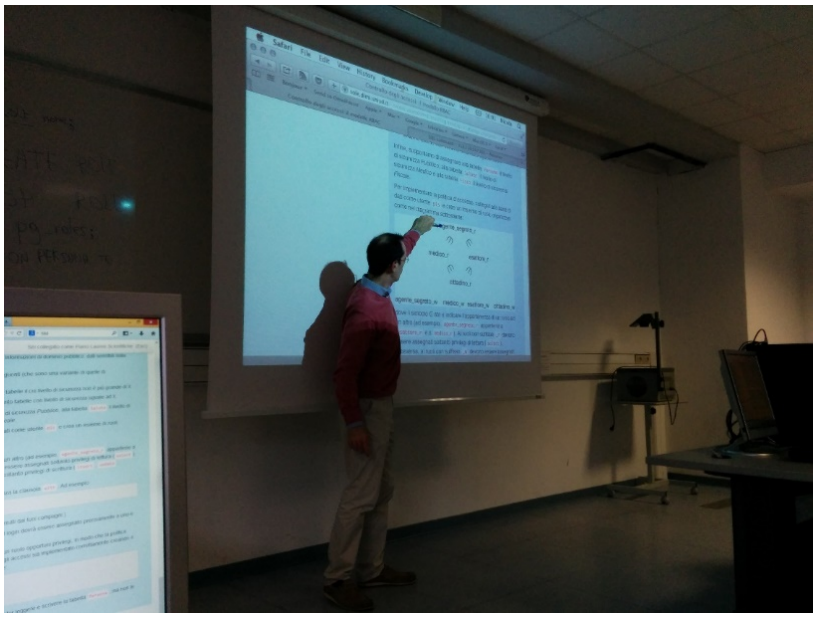
- **Consenso:** l'informazione dev'essere memorizzata con il con-


Spie, cavalli di Troia e basi di dati ippocratiche
Nicola Vitacolonna
Dipartimento di Matematica e Informatica
Università degli Studi di Udine
4 aprile 2014


SQL Injection
• Tecnica principale per l'intrusione in una base di dati
• OWASP
Codice dell'applicazione:
Input dell'utente:
1 or (0 = 0); drop table Professore;
SQL inviato al DBMS server:
drop table Professore;


Login


Per implementare la politica d'accesso, collegati alla base di
dati come utente pls e crea un insieme di ruoli, organizzati
come nel diagramma sottostante:
agente_segreto_r
medico_r
esattore_r
cittadino_r
agente_segreto_w medico_w esattore_w cittadino_w
dove il simbolo ∈ sta a indicare l'appartenenza di un ruolo ad
un altro (ad esempio, agente_segreto_r appartiene a
esattore_r e a medico_r). Ai ruoli con suffisso _r devono
essere assegnati soltanto privilegi di lettura (select).
Viceversa, ai ruoli con suffisso _w devono essere assegnati
soltanto privilegi di scrittura (insert , update , delete).
Questi ruoli non sono usati per il login, perciò vanno creati
senza la clausola with . Ad esempio:
create role nome_cittadino_r;



senso di chi la fornisce;

- **Contenuti limitati:** solo i dati personali strettamente necessari per ottenere gli scopi previsti devono essere memorizzati nella base di dati;
- **Uso limitato:** solo le interrogazioni consistenti con gli scopi previsti possono essere eseguite;
- **Diffusione limitata:** le informazioni personali nella base di dati non devono essere diffuse senza il consenso di chi le fornisce;
- **Durata limitata:** le informazioni devono essere mantenute nella base di dati solo per il tempo strettamente necessario al raggiungimento degli scopi previsti;
- **Accuratezza:** le informazioni personali memorizzate nella base di dati devono essere aggiornate e accurate;
- **Sicurezza:** le informazioni personali devono essere protette da usi inappropriati e accessi non autorizzati;
- **Apertura:** i soggetti i cui dati sono memorizzati nella base di dati devono poter accedere a tutte le informazioni che li riguardano;
- **Conformità:** i soggetti i cui dati sono memorizzati nella base di dati devono poter verificare il rispetto dei suddetti principi.

In alcuni di questi requisiti si riconosceranno richiami alla nostra legislazione sulla privacy. Ma sono davvero tutti principi realizzati? Saprà la futura generazione di ingegneri, tecnici e informatici tenerne conto?

8.6 Ringraziamenti

Si ringrazia la prof.ssa Mariagemma Fantin per aver contribuito al buon esito del progetto.

8.7 Istanza usata per l'esercizio della Sezione 8.3

Nome	sesso	professione	contributo
Aldo	M	giornalista	3000.00
Biagio	M	giornalista	500.00
Carlo	M	imprenditore	1.00
Daria	F	giornalista	5000.00
Elena	F	professore	1000.00
Fabio	M	professore	20000.00
Giulia	F	medico	2000.00
Ivano	M	avvocato	10000.00

Riferimenti bibliografici

[2] Rakesh Agrawal et al. «Hippocratic Databases». In: *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment. 2002, pp. 143–154.

[8] David Elliott Bell. «Looking Back at the Bell-La Padula Model». In: *ACSAC*. Vol. 5. 2005, pp. 337–351.

[9] David Elliott Bell e Leonard J. LaPadula. *Secure computer systems: Mathematical foundations*. Rapp. tecn. DTIC Document, 1973.

[12] Elisa Bertino e Ravi Sandhu. «Database security—concepts, approaches, and challenges». In: *IEEE Transactions on Dependable and Secure Computing* 2.1 (2005), pp. 2–19.

[32] Edgar F. Codd. «A Relational Model of Data for Large Shared Data Banks». In: *Information Retrieval* 13.6 (giu. 1970), pp. 377–387.

[33] Edgar F. Codd. «Data Models in Database Management». In: *Proceedings of the 1980 Workshop on Data Abstraction, Databases and Conceptual Modeling*. 1980, pp. 112–114.

[34] Edgar F Codd. *Derivability, redundancy and consistency of relations stored in large data banks.* Rapp. tecn. Reprinted in SIGMOD Record, March 2009 (Vol. 38, No. 1). IBM, 1969, pp. 17–36.

[35] Edgar F. Codd. *The Relational Model for Database Management: Version 2*. Addison-Wesley, 1990.

[39] Chris J. Date. *An Introduction to Database Systems*. 8th. Boston: Pearson/Addison Wesley, 2004. ISBN: 0321197844.

[43] Dorothy E. Denning, Peter J. Denning e Mayer D. Schwartz. «The Tracker: A Threat to Statistical Database Security». In: *ACM Transactions on Database Systems (TODS)* 4.1 (1979), pp. 76–96.

[44] Dorothy Elizabeth Robling Denning e Peter J. Denning. «Data Security». In: *Computing Surveys* II.3 (set. 1979), pp. 227–249.

[49] Cynthia Dwork. «A Firm Foundation for Private Data Analysis». In: *Communications of the ACM* 54.1 (2011), pp. 86–95.

[50] Cynthia Dwork. «Differential Privacy: A Survey of Results». In: *Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[53] Ramez Elmasri e Shamkant Navathe. *Fundamentals of database systems*. 6th. Addison Wesley, apr. 2010.

[85] Sylvia Osborn, Ravi Sandhu e Qamar Munawer. «Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies». In: *ACM Transactions on Information and System Security (TISSEC)* 3.2 (2000), pp. 85–106.

[93] Dorothy Elizabeth Robling Denning. *Cryptography and Data Security*. Addison-Wesley Longman Publishing Co., Inc., 1982.

[95] Ravi S. Sandhu. «Lattice-Based Access Control Models». In: *Computer* 26.11 (1993), pp. 9–19.

[98] Bruce Schneier. *The Eternal Value of Privacy*. https://www.schneier.com/essay-114.html. 2006.

[99] Bruce Schneier. *The US government has betrayed the internet. We need to take it back*. The Guardian, 13 settembre 2013. http://www.theguardian.com/commentisfree/2013/sep/05/government-betrayed-internet-nsa-spying. URL: http://www.theguardian.com/commentisfree/2013/sep/05/government-betrayed-internet-nsa-spying.

[113] Alan Westin. *Freebies and Privacy: What Net Users Think*. Rapp. tecn. Technical Report, Opinion Research Corporation, 1999.