Controllo degli accessi e autenticazione

Controllo degli accessi

Politiche di sicurezza

Durante la pianificazione della sicurezza, nell'ambito prevenzione dobbiamo specificare:

- Da chi o da cosa ci si vuole proteggere;
- Quali proprieta' di sicurezza devono essere soddisfatte dal sistema;
- Quali sono gli utenti autorizzati e quali non;
- Che livello di segretezza hanno le risorse (top secret, secret, riservato, non classificato).

In particolare si devono:

- Definire delle **regole di accesso** e controllare che chi accedere alle risorse soddisfi tali regole;
- Verificare che il **sistema funzioni come previsto**;
- Trasmetterei i **dati non in chiaro**.

Regole di accesso

L'obiettivo del modulo di controllo degli accessi e' garantire che gli utenti abbiano accesso a tutte e solo le risorse e i servizi ai quali sono autorizzati, per ottenere cio' bisogna:

- Definire una **politica di sicurezza** che limiti quali operazioni gli utenti possono fare sulle risorse;
- Scegliere un **meccanismo** per far rispettare la politica (ossia implementare la politica).
- Verificare che sia la politica che il meccanismo siano validi e coerenti.

Politiche e meccanismi per il controllo degli accessi si differenziano per il livello di astrazione. Ragionare su diversi livelli di astrazione permette di mantere separata la fase di implementazione da quella di analisi e specifica dei requisiti; inoltre permette il confronto tra diverse politiche per il controllo degli accessi e il confronto tra diversi meccanismi che implementano la stessa politica.

E' possibile realizzare meccanismi che implementano piu' politiche ma se le politiche sono in conflitto, le discrepanze possono diventare una vulnerabilita'.

Politiche

Sono regole ad alto livello che descrivono gli accessi autorizzati al sistema:

- Un insieme di proprieta' di sicurezza;
- Dipendono dall'applicazione da proteggere.

Esempio in una banca:

- Autenticazione dei clienti agli sportelli, ATM e sul web;
- Non-repudiation delle transazioni;
- Integrita' dei conti correnti dei clienti;
- Segretezza dei clienti e dei dati interni;
- Disponibilita' di un sistema di allarme;
- Separation of duties (nessun conflitto di interessi).

Le politiche di sicurezza possono essere descritte con:

- Linguaggio naturale: semplice da capire ma di solito impreciso;

- Matematica: precisa ma difficile da capire;
- Linguaggio ad-hoc: pseudo codice che cerca di bilanciare la precisione con la facilita' di comprensione.

Meccanismi

Sono funzioni di basso livello (HW e SW che implementano le politiche).

Terminologia per il controllo degli accessi

Gli elementi principali sono:

- **Soggetti**: le entita' attive, come utenti e processi che richiedono l'accesso ad una certa risorsa o un servizio;
- **Oggetti**: le entita' passive, come risorse, file, servizi;
- **Operazioni di accesso (o modalita' di accesso)**: come un soggetto puo' accedere ad un oggetto (varia a seconda delle tipologie di oggetti che i soggetti richiedono);
- **Permesso (o diritto di accesso)**: possibilita' di accedere ad una risorse secondo una certa modalita'/operazione di accesso;
- **Privilegio**: un insieme di permessi dati direttamente ad un ruolo specifico (es. amministratore, operatore, ecc.).

A livello elementare le modalita' di accesso di un soggetto su un oggetto sono:

- **Osservare** un oggetto;
- **Alterare** un oggetto;

Esempio di modalita' di accesso su Unix/Linux

Le modalita' di accesso ai file sono:

- **read**: lettura di file;
- **write**: scrittura su file;
- **execute**: esecuzione di un file eseguibile.

Le modalita' di accesso alle directory sono:

- **read**: elencare gli elementi della directory;
- **write**: creare o rinominare file e subdirectory in una directory;
- **execute**: ricerca in una directory.

Funzionamento di un sistema per il controllo degli accessi

Un sistema di controllo degli accessi regola le operazioni che possono venire eseguite sui dati e sulle risorse che devono essere protette. L'obiettivo e' quello di controllare le operazioni eseguite dai soggetti per prevenire azioni che potrebbero danneggiare i dati o le risorse (o violare delle proprieta' di sicurezza che sono definite nella politica di sicurezza).

Il controllo in genere viene fatto dal sistema operativo e in particolare dal **reference monitor**: una macchina astratta che applica il controllo degli accessi tramite l'intercettazione delle richieste d'accesso e la fornitura di una risposta alle richieste provenienti dai soggetti.

Nel reference monitor:

- Un soggetto attivo chiede accesso ad un oggetto passsivo per eseguire una specifica operazione di accesso;
- Il monitor riceve la richiesta, consulta la polita di sicurezza che deve applicare e concede o meno l'accesso.

La correttezza dell'intero sistema di controllo degli accessi implica:

- Una corretta identificazione/autenticazione del soggetto attivo;
- Una corretta definizione e implementazione della politica di sicurezza.

Vediamo come il concetto di autenticazione e' strettamente correlato a quello di autorizzazione, ma hanno definizioni molto diverse:

- **Autenticazione**: il reference monitor verifica l'identita' del soggetto attivo che fa la richiesta;

- **Autorizzazione**: il reference monitor decide se concedere o meno l'accesso.

Politiche per il controllo degli accessi

La responsabilita' di stabilire una politica puo' venire assegnata:

- Al **proprietario della risorsa**, che puo' decidere a chi e' permesso l'accesso:
  - In questo caso si parla di **politiche discrezionarie (DAC)** in quanto il controllo degli accessi e' a discrezione del proprietario.
- Ad una politica **a livello di sistema** che stabilisce gli accessi:
  - In questo caso, poiche' il sistema e' centralizzato, tali politiche si definiscono **mandatorie (MAC)**.

Esistono altre interpretazioni di Discretionary Access Control e Mandatory Access Control, che pero' ormai sono in disuso.

Discretionary Access Control

Il proprietario della risorsa:

- Decide gli accessi: chi puo' accedere e in quale modalita';
- Ha autorita' di **passare i propri privilegi** ad altri utenti: **delegation of duty**.

Il limite di questa politica e' il fatto che controlla solo gli accessi diretti alla risorsa: non c'è alcun controllo su cosa accade alla risorsa dopo che vi e' stato concesso l'accesso ad un soggetto. Cio' predispone la politica ed essere particolarmente vunerabile ai **Trojan Horse** che contengono codice nascosto malevolo per compiere azioni illegittime.

Esempio vulnerabilita' ai Trojan Horse: Il soggetto S1 ha accesso in lettura al file F1 e in scrittura al file F2. Il soggetto S2 ha accesso in lettura al file F2.

Se il soggetto S1 esegue una funzione che:

- Legge da F1;
- Scrive su F2;

La politica DAC viene rispettata, ma la funziona scrivera' il contenuto di F1 su F2, rendendo possibile a S2 la lettura del file F1.

Mandatory Access Control

La politica mandatoria e' una politca **multilivello** gestisce in modo centralizzato gli accessi. In particolare assegna livelli di sicurezza a soggetti ed oggetti senza che i soggetti abbiano alcun controllo sul livello di sicurezza che gli viene assegnato. La modalita' di accesso e' un attributo dell'ogetto e impone delle restrizioni sul flusso dell'informazione, per questo motivo non sono possibili attacchi di tipo Trojan Horse.

Esistono due tipi di politiche multilivello a seconda delle proprieta' di sicurezza che si vogliono garantire:

- **Bell - La Padula**: secrecy based;
- **Biba**: integrity based.

Modello Bell - La Padula

E' un modello usato per garantire la segretezza in cui soggetti ed oggetti vengono classificati in base al livello di sicurezza:

- Il livello del soggetto indica il livello di autorizzazione/fiducia associato all'utente;
- Il livello dell'oggetto rappresenta la sensibilita' dell'informazione.

I livelli possono formare un reticolo e possono essere di tipo:

- Trusted/Untrusted
- Public/Secret/Top Secret

Le regole della politica mandatoria che si basa su questo modello sono:

- **No write-down**: un soggetto di alto livello non puo' scrivere oggetti di livello inferiore (previene il downgrading dell'informazione);
- **No read-up**: un soggetto di livello inferiore non puo' accedere ad un oggetto di livello superiore (garantisce la segretezza).

Modello Biba

E' un modello usato per garantire l'integrita', per questo, soggetti ed oggetti vengono classificati per livello di integrita':

- Il livello del soggetto indica il livello di fiducia associato all'utente;
- Il livello dell'oggetto rappresenta la fiducia nella validita' dell'informazione.

Le regole sono:

- **No write-up**: non e' possibile scrivere informazioni prelevate da un oggetto di basso livello su un oggetto di piu' alto livello (dati integri potrebbero venire corrotti);
- **No read-down**: un soggetto di alto livello non puo' leggere oggetti di livello inferiore (i dati potrebbero non essere validi).

Role-Based Access Control

E' basato sull'idea che uno stesso soggetto puo' aver bisogno di permessi diversi a seconda dell'attivita' (ruolo) che svolge. In questo tipo di politica l'accesso agli oggetti e' mediato dai ruoli: un soggetto s con ruolo r ha tutti i permessi associati al ruolo r.

I vantaggi di questo approccio sono:

- **Least Privilege**: ogni soggetto, nel momento in cui compie un'azione, ha il privilegio minimo per compiere tale azione;
- I permessi sono facilmente **revocabili**;
- **Separation of duty**: tramite la creazione di ruoli diversi in base all'azione che deve svolgere il soggetto;
- **Gerarchie di ruoli**;
- **Anonimity** (parziale): in quanto non e' il soggetto a compiere l'azione ma e' il soggetto con quel ruolo specifico che la sta compiendo.

Il **gruppo** e' un insieme di utenti. Il **ruolo** e' un insieme dinamico di permessi/privilegi.

Memorizzazione della matrice degli accessi

Dobbiamo tener conto anche degli aspetti implementativi quando parliamo di politiche degli accessi, e la memorizzazione di una matrice degli accessi e' spesso un tema da affrontare.

In genere, quando si realizza una matrice degli accessi, si mettono i soggetti sulle righe e gli oggetti sulle colonne, nella cella si specifica il privilegio che il determinato soggetto ha su quello specifico oggetto:

> Accesso: Soggetto X Oggetto -> Permesso

La matrice risultante e' grande e sparsa, quindi non e' efficiente memorizzarla in questo modo. In genere si memorizza:

- Per colonne: lista di soggetti che puo' accedere o meno a una data risorsa (**ACL - Access Control List**);
- Per righe: lista di risorse a cui puo' accedere un soggetto (**Capability List**);
- Solo le triple non-nulle: che e' il meccanismo utilizzato dai database (**Tabelle di autorizzazione**).

Access Control List

Elenca i soggetti che possono accedere all'oggetto e con quale privilegio. Si tratta di una memorizzazione piu' compatta rispetto alla matrice degli accessi, permette di avere un sommario semplice da leggere, ma e' difficile revocare i privilegi su un soggetto perche' si devono scorrere tutte le Access Control Lists per vedere se e' presente un soggetto e in caso rimuoverlo o modificarne i privilegi.

E' una memorizzazione molto utile in caso di pochi soggetti e molti oggetti.

Capability List

Elenca gli oggetti ai quali il soggetto puo' accedere e con quale privilegio. E' una memorizzazione compatta e particolarmente utile quando si vuole delegare a qualcuno i propri privilegi, ma scomoda per revocare una capability, in quanto si devono scorrere tutte le Capability Lists ed effettuare la modifica o rimozione.

Controllo degli accessi in Linux

I sistemi operativi UNIX fanno uso della politica DAC, dove:

- I soggetti sono gli utenti (root compreso);
- Ogni file ha un **proprietario** e un **gruppo**:
  - Le modalita' di accesso sono: **r**ead, **w**rite ed e**x**ecute;
  - Tutti i file sono protetti da ACLs:
    - Indicano le modalita' di accesso per user, group e others;
    - I privilegi sono indicati mediante 9 bit.
- I programmi vengono eseguiti in un'area di memoria protetta:
  - Vengono eseguiti tramite i privilegi del chiamante tranne in caso di suid/sgid.

Autenticazione uomo-macchina

Il problema dell'autenticazione

Il reference monitor si comporta correttamente sia se implementa correttamente la politica, che se fa precedere l'autenticazione all'autorizzazione di un soggetto verso un oggetto.

Nella vita reale, una forma di autenticazione viene sempre richiesta prima di compiere un'azione (ad esempio presentando la Carta d'Identita' prima di poter incassare un'assegno). Cosi' anche nei sistemi automatici, il principio e' lo stesso e si puo' fare in due modi:

- **Autenticazione prima dell'autorizzazione**;
- L'utente effettua una **registrazione** che viene salvata nei logo di sistema, per tenere traccia degli utenti che hanno richiesto un certo servizio.

Definizione di autenticazione:

> Processo di verifica dell'identita' di un utente.

In particolare si parla di **mutua autenticazione** quando si deve stabilire l'identita' di **entrambe le parti** che interagiscono. Le parti possono essere utenti, computer e anche processi.

Nel caso dell'autenticazione utente-computer gli elementi in gioco sono:

- L'**utente** che si deve autenticare;
- La **caratteristica** che contraddistingue la utente:
  - Qualcosa che l'utente **conosce**: pin, password, etc.
  - Qualcosa che l'utente **possiede fisicamente**: chiavi, carte magnetiche, smart card, etc.
  - Qualcosa che l'utente **e'**: impronte digitali, iride, tono della voce, etc.
- L'**amministratore** del sistema;
- Il meccanismo di **autenticazione**:
  - Basandosi sulla coppia identita' dell'utente:caratteristica dell'utente;
- Il meccanismo di **autorizzazione**.

Autenticazione basata sulla conoscenza

Il funzionamento e' molto semplice: l'utente immette lo username per l'identificazione e la password per l'autenticazione e il sistema controlla la password per autenticare l'utente.

Proprio per questa semplicita' e' il sistema piu' diffuso, oltre ad essere economico e facilmente implementabile. Purtroppo pero' e' anche il sistema piu' debole.

I problemi che insorgono sono:

- Il sistema **come e dove memorizza** le passwords?
- Il sistema **come verifica** le passwords?
- Quanto e' semplice **indovinare** una password?

Password memorizzate in chiaro

Le password vengono memorizzate in chiaro su un file protetto da un meccanismo di controllo dell'accesso e la validazione avviene confrontanto la password inserita con quella memorizzata.

Questo metodo fu utilizzato negli anni '60 da uno dei primi sistemi operativi multi-utente, tuttavia presenta delle evidenti problematiche qualora qualcuno si impossessasse del file, in quanto con esso potra' impersonare qualsiasi utente.

In particolare l'autorizzazione deve dipendere dall'autenticazione, ma in questo caso **la fase dell'autenticazione dipende dall'autorizzazione** (e' il contrario) in quanto si deve stabilire chi ha accesso al file delle password (il che e' un controsenso).

Password memorizzate cifrate

Un approccio piu' sicuro implica la cifratura dell'intero file o solo delle singole password. Per la verifica si puo' procedere in due modi:

- Il sistema decifra la password memorizzata e la confronta con quella inserita dall'utente;
- Il sistema cifra la password inserita dall'utente e la confronta con quella cifrata:
  - In questo caso la cifratura puo' avvenire tramite metodi crittografici o tramite un algoritmo di hashing (irreversibile).

Con questo metodo di memorizzazione delle password il problema e' che **due utenti con la stessa password hanno entry identiche nel file delle password**, il che rende l'attacco di ricerca piu' veloce.

Su Unix/Linux

In passato le password venivano memorizzate crittate con l'algoritmo di cifratura simmetrico 25xDES, piu' recentemente si e' adottato l'hash tramite MD5.

In particolare viene applicata la funzione di hashing alla stringa composta dalla concatenazione di:

- Una sequenza di 48-128 bit generata dal sistema detta **salt** (unica per ogni utente);
- La password.

La tabella delle password risultante e':

username	salt	hash(salt, password)
user1	abcd	hash(abcd, password1)

Il salt viene memorizzato in chiaro perche' e' necessario conoscerlo nella fase di verifica della password. Il sistema, dopo aver ricevuto le credenziali per l'autenticazione, recupera il salt dell'utente, lo concatena con la password ricevuta e applica l'algoritmo di hash per confrontarlo con quello memorizzato.

Il salt offre diversi vantaggi:

- Due diversi utenti con la stessa password avranno hash diverso:
  - In quanto il salt si basa sulla data e l'ora in cui e' stato creato l'utente;
- La lunghezza delle password da cifrare e' maggiore per via dei bit in piu':
  - Un attacco bruteforce ci mettera' un tempo esponenzialmente piu' lungo per ottenere una password.

Attacco bruteforce (o attacco esaustivo)

L'attaccante prova in modo sistematico tutte le possibili password, ovvero tutte le possibili stringhe che comprendono tutti i caratteri ammessi.

Attacco dizionario

L'attaccante prova le password piu' probabili basandosi sul fatto che gli utenti spesso preferiscono password corte e legano la propria password a qualcosa che abbia un particolare significato per loro (es. nome, data di nascita, squadra di calcio, ectc.)

Caratteristiche delle password

Vulnerabilita'

Le password rischiano di:

- Essere indovinate: **guessing**;
- Essere sbirciate mentre vengono inserite: **snooping** o **shoulder surfing**;
- Essere intercettate tramite trasmissione in rete: **sniffing** e **keystroke sniffing**;
- Venire aquisite da terze parti che impersonificano l'interfaccia di login: **trojan login** o **spoofing**;
- Essere "rubate" tramite **social engineering**.

Difese

Per evitare i **Guessing Attacks**:

- Possiamo verificare i log del sistema per verificare se e' stato tentato l'accesso;
- Mettiamo un limite al numero di sbagli permessi;
- Usiamo una password con caratteri non alfanumerici che rendono gli attacchi dizionario molto piu' difficili.

Per evitare i **Sniffing Attacks**:

- Schermiamo la password scritta (*******);
- Proteggiamo l'area di memoria in cui vengono salvati temporaneamente i keystrokes (per il keystroke sniffing).

Per evitare gli **Offline dictionary attacks**:

- Facciamo in modo che il file delle password sia accessibile solo da utenti con privilegi da amministratore (Shadow password di UNIX).

Cause di vulnerabilita'

Le cause principali delle vulnerabilita' delle password sono note:

- Password immutata per lungo tempo;
- Condivisione con amici/colleghi;
- Scelta di password debili;
- Stessa password su piu' computer/account;
- Scrittura delle password su supporto fisico (carta, lavagna, etc).

Buona gestione

Partendo dalle cause principali delle vulnerabilita' possiamo stilare una lista di regole per una buona gestione delle password:

- Cambiare password frequentemente;
- Non condividere le password con nessuno;
- Non usare la stessa password per autenticazioni diverse;
- Usare almeno 8 caratteri;
- Non usare una parola del dizionario;

- Bilanciare tra semplicita' di memorizzazione e complessita':
  - Deve essere facile da ricordare, non serve trascriverla;
  - Difficile da intuire, protezione contro il password guessing.

Controlli automatici

Per obbligare l'utente a seguire almeno alcune linee guida per la gestione delle password, in generale il sistema effettua controlli automatici quando l'utente inserisce per la prima volta la password:

- Restrizioni sulla lunghezza e sul numero minimo di caratteri;
- Combinazioni con caratteri alfanumerici;
- Controllo rispetto ai dizionari:
  - Rifiuto delle parole del linguaggio naturale;
- Verifica del tempo massimo di validita':
  - L'utente deve cambiare la password quando scade.

Distribuzione iniziale

Durante la fase creazione dell'account, e' necessario che l'utente possa accedere per la prima volta:

- L'utente si reca dall'amministratore per richiedere la password:
  - Scomodo per l'utente;
  - Pericoloso per il sistema;
- No password o password di default:
  - Pericoloso qualora non si accede tempestivamente per il primo inserimento inserimento o la modifica della password di default;
- Password spedita via posta/mail:
  - Il messaggio puo' venire intercettato;
- L'amministratore prepare l'account con una password iniziale gia' scaduta(pre-expired password):
  - L'utente e' obbligato a modificarla entro un tempo massimo;
  - Controllo della password scelta (lunghezza, caratteri, etc.).

Autenticazione basata sul possesso

La prova dell'identita' viene fornita dal possesso di un **token** che puo' essere:

- Carta magnetica;
- Smart card:
  - Memory card: con memoria ma non capacita' computazionali;
  - Microprocessor card: con memoria e microprocessore;
- Smart token: generatori di One Time Passwords (OTP);
- Tag RFID (Radio Frequency IDentification).

Indipendentemente dal token usato, lo svantaggio principale di questo metodo di autenticazione e' che si verifica **l'identita' del token e non quella dell'utente**, infatti rubando o falsificando un token si puo' impersonificare l'utente. E' pero' difficile estrarre un segreto da un token, quindi la soluzione piu' diffusa e' usare una combinazione tra l'autenticazione basata sulla conoscenza e sul possesso (Bancomat: carta + pin).

Carta magnetica

Ha una striscia dove si possono memorizzare poche informazioni (in genere 250 byte) e puo' facilmente venire clonata (forging). Durante la fase di autenticazione il pin puo' venire verificato:

- Da un server/autenticatore online centralizzato
- Confrontando il pin immesso con quello salvato sulla carta

Smart card

Le smart cards possono contenere un microprocessore (a 8, 16, 32 bits), ha piu' memoria della carta magnetica (RAM, ROM, EEPROM, Flash), ha un canale di I/O (contact o contact-less) e ha anche una processore crittografico per essere utilizzata per meccanismi di tipo challenge-response.

Viene usata come carta di credito, bancomat, SIM - Subscriber Identity Module che memorizza l'identita' del cliente e il suo PIN, nell'ambito delle pay-tv, badge, etc.

Smart token

E' un dispositivo che oltre ad avere un microprocessore e una memoria, ha anche un display e a volte una tastierina (un vero e proprio computer). In genere viene utilizzato come generatore di OTP o come contenitore di credenziali principali.

Gli svantaggi nell'utilizzo degli smart token risiedono principalmente nella gestione durante periodi transitori in caso di furto o smarrimento. Inoltre adottare smart tokens porta dei costi aggiuntivi per la progettazione.

Funzionamento:

- Una chiave segreta (seed - seme) viene memorizzata all'interno dello smart token dal produttore e viene condivisa con il server;
- Quando lo smart token deve generare l'OTP prende il seme e alcune informazioni esterne (PIN, data, ora, etc.);
- La password viene visualizzata sul display per 30-90 secondi, e trascorso questo intervallo di tempo genera una nuova password;
- La sincronizzazione con il server avviene grazie al fatto che il seme e' condiviso, l'algoritmo e' noto a entrambi e l'orologio del server e dello smart token sono sincronizzati.

RSA SecurID

E' un tipo di smart token utilizzato da molte banche e fa uso di un'autenticazione a due livelli:

- Pin segreto (password tramite conoscenza): scelto dall'utente e condiviso con il server;
- Codice generato dal token (password tramite possesso): dentro al token c'e' un orologio sincronizzato con il server e e aintervalli regolari (o se si accende il display) il tempo e la chiave segreta vengono utilizzati e combinati per generare una password che scade in 60 secondi.

eToken PRO

E' uno smart token USB che contiene delle credenziali digitali. L'utente inserisce l'eToekn PRO nella porta USB, digita la propria password e le chiavi contenute nello smart token sono a disposizione.

Tag RFID

E' una particolare etichetta elettronica che serve a memorizzare informazioni relative all'oggetto o animale a cui e' attaccata. E' costituita da 3 elementi fondamentali:

- Un apparecchio di lettura/scrittura;
- Una o piu' etichette RFID (chiamate tag o Transponder);
- Un sistema informativo di gestione dei dati per il trasferimento degli stessi da e verso i lettori.

L'etichetta RFID puo' essere attiva, passiva, semi-attiva o semi-passiva: a seconda se si tratta di un apparecchio di sola lettura o anche scrittura. Se e' attiva dispone di una batteria per l'alimentazione, una o piu' antenne per inviare e ricevere segnali, uno o piu' tag.

Autenticazione basata su caratteristiche biometriche

La prova dell'identita' e' fornita dal possesso di alcune caratteristiche univoche:

- **Fisiche**: impronta digitale, della retina, dell'iride, del viso, etc;
- **Comportamentali**: firma, timbro della voce, scrittura, keystroke dynamic, etc.

Per scegliere una caratteristica da utilizzare bisogna trovarne una che abbia le seguenti proprieta':

- **Universalita'**: tutte le persone dovrebbero possederla;
- **Unicita'**: ogni persona l'ha distinta dalle altre;
- **Stabilita'**: non deve cambiare nel tempo e non puo' venire modificata;
- **Facilita' di rilevamento**;
- **Accettabilita'** (da parte dell'utente, quindi non troppo intrusiva);
- **Difficolta' di contraffazione**.

L'utilizzo delle caratteristiche biometriche e' una soluzione promettente che risolve del tutto il problema dell'impersonificazione, in quanto non puo' venire persa, rubata o indovinata.

Purtroppo e' tecnicamente meno accurata, costosa e intrusiva, quindi non sempre accettata dagli utenti.

L'autenticazione basata su caratteristiche biometriche necessita di una fase di **campionamento** tramite la definizione di un **template**: ossia la rappresentazione digitale delle caratteristiche univoche del dato biometrico scelto. In particolare, durante la fase di campionamento, vengono fatte piu' misurazioni della caratteristica interessata (in quanto la perfetta uguaglianza tra due template e' praticamente impossibile).

Durante l'autenticazione si confronta la caratteristica appena misurata rispetto al template, la quale avra' successo se le due rilevazioni corrispondono **a meno di una tolleranza** che va definita attentamente a priori.

> La perfetta uguaglianza tra due campioni dello stesso utente e' praticamente impossibile, dunque il problema risiede nel confrontare la caratteristica appena misurata con il template di quell'utente, distinguendola dal template di un altro utente.

Alcune metriche della biometria

FAR - False Acceptance Rate

Probabilita' che il sistema accetti un impostore: ovvero che il sistema associ un campione in input con il template relativo all'utente sbagliato.

FRR - False Reject Rate

Probabilita' che il sistema non riconosca un utente valido: ovvero che il sistema non associ il campione in input con il template dello stesso utente.

Soglia di tolleranza

Il FAR e l'FRR tornano utili quando si deve stabilire la soglia di tolleranza dell'errore, ovvero quale essere la differenza tra il valore misurato e il valore template:

- Se la soglia e' troppo alta: si accettano gli impostori (FAR alto);
- Se la soglia e' troppo bassa: non si accettano utenti legittimi (FRR alto).

Vanno dunque bilanciate le due metriche, per ottenere l'**ERR - Equal Error Rate**.

Tipologie di attacchi

L'attaccante puo':

- Usare una caratteristica biometrica contraffatta (guanto, lente a contatto, maschera);
- Riutilizzare un vecchio template;
- Intercettare la comunicazione tra il sensore e il database dove e' memorizzatoi il template, sovrascrivendo la decisione finale;
- Modificare il template nel database.

Caratteristiche biometriche

Devono soddisfare le proprieta' descritte prima.

Impronte digitali

E' uno dei metodo piu comuni e affidabili per il riconoscimento d'identita', in passato tramite l'inchiostro, da qualche anno con i lettori ottici:

- Fa riferimento alle piccole righe che si formano su mani e piedi ancor prima della nascita;
- Restano inalterate per tutta la vita dell'individuo (a meno di incidenti);

Digitalizzazione dell'impronta

Le impronte sono classificate in 3 macro-gruppi in base allo schema predominante:

- Arch Schema (5% della popolazione);
- Whorl Schema (35% della popolazione);
- Loop Schema (60% della popolazione);
- Accidental Schema (combinazione di piu' schemi).

Il campione digitale dell'impronta del dito rappresenta le **minuzie** (dei punti di biforcazione o di terminazione delle linee) presenti nell'impronta, che sono uniche per ogni individuo. In fase di autenticazione saranno confrontati i punti di minuzie.

In particolare implementare l'autenticazione tramite fingerprint e' economico, di piccole dimensioni e low power, gli unici svantaggi sono rappresentati da ferite e cicatrici che gli utenti possono avere (che possono falsare il campione) e da persone con poche minuzie.

Forma della mano

Tramite uno scanner e' possibile memorizzare molte informazioni riguardo la forma della mano di un soggetto, come la lunghezza delle dita, l'ampiezza del palmo della mano, etc.

E' una tecnica piu' affidabile delle impronte digitali e i template generati sono molto piu' piccoli (10 byte vs 250-1000 byte).

Pero' richiede scanner molto grandi e costosi ed e' difficile gestire le frequenti ferite alle mani che gli utenti possono avere (la proprieta' di stabilita' non e' sempre garantita).

Riconoscimento facciale

Richiede un'espressione neutrale perche' il campione digitale si basa sulle distanze tra occhi, naso, bocca, etc.

E' una rilevazione totalmente non intrusiva in quanto puo' avvenire anche a una certa distanza, ma alcune persone possono non accettare il fatto che venga fatta una foto. Inoltre i valori FRR e FAR possono fluttuare molto per via dell'espressione, dunque e' difficile definire una soglia di tolleranza.