User Authentication

L'autenticazione è un’unione tra **un'identità** e un **soggetto**, ovvero è il processo di verifica dell'identità dichiarata da un soggetto. L’autenticazione è composta da due fasi principali:

1. Fase di **identificazione** → in cui viene presentato un identificatore al sistema di sicurezza;
2. Fase di **verifica** → in cui si presentano o si generano le informazioni di autenticazione, che confermano il legame che vi è tra l’entitià e l’identificatore (ad esempio la password).

Riassumendo questi primi concetti, quindi, possiamo dire che:

- l’identificatore è il mezzo con cui un utente fornisce una determinata identità (ovvero una identità rivendicata) al sistema;
- l’autenticazione è il mezzo mediante il quale si stabilisce la validità della rivendicazione;
- se nessuno è in grado di ottenere o indovinare la password di un utente, allora la combinazione dello user-ID e della password permette agli amministratori del sistema di impostare i permessi dell’utente e di verificarne le sue attività.

Quando parliamo di autenticazione, vi sono quattro possibili mezzi per autenticarsi:

1. Qualcosa che l’individuo/l’utente **conosce** → come per esempio: password, PIN, risposte ad un set di domande predefinite;
2. Qualcosa che l’individuo **possiede** → come per esempio: badge o smart card;
3. Qualcosa che l’individuo **è** (biometria statica)→ come per esempio: impronta digitale, retina, viso, voce;
4. Qualcosa che l’individuo **fa** (biometria dinamica) → come per esempio: modello di voce, caratteristiche della scrittura a mano, ritmo della battitura sulla tastiera.

Sorge a questo punto spontanea la domanda: Come mai l’autenticazione è importante? L’autenticazione è importante per due motivi:

- **Accountability** → ovverosia tenere traccia delle azioni compiute dall’utente tramite i file di log. Quindi, l’autenticazione diventa utile per tutto il processo di identificazione delle responsabilità → per avere un’identità serve che venga assegnata un’identità agli utenti che agiscono nel sistema;

- **Regolamentare il controllo degli accessi** → ovvero data l'identità dell'utente dobbiamo stabilire cosa può e non può fare (come per esempio, nel sito dell'Ateneo, gli studenti hanno minori diritti di accesso rispetto ai professori).

Il metodo più utilizzato per l'autenticazione è l'utilizzo della **password**, a cui siamo ampiamente abituati (ovvero non ci dà fastidio, a livello di usabilità, autenticarci con una password ad un sito). Tale metodo funziona nel seguente modo:

- un utente inserisce il proprio username e password;
- il Server a cui arriva la richiesta, andrà a confrontare e in caso positivo a matchare la password con quella contenuta nel file delle password → nel caso in cui le password combacino, allora l'utente avrà accesso al sistema.

La password è una stringa alfa-numerica associata ad un utente, nota solamente all'utente e al sistema, che conferma l'identità del soggetto. La password più semplice è una sequenza di caratteri. Lo spazio delle password è l'insieme di tutte le sequenze di caratteri che possono essere password. La password può essere combinata con altre **informazioni complementari** (ovvero un insieme di informazioni, che il sistema memorizza per convalidare le informazioni di autenticazione → quindi per informazioni complementari intendiamo informazioni legate al contento, che possono essere utilizzato insieme alle password). La protezione tramite password sembra sicura, tuttavia **i comuni errori umani** riducono la sicurezza dei sistemi basati su password → le password rimangono uno dei principali punti di accesso degli attacchi, in quanto molte password sono facilmente rompibili → questo porta al fatto, che gli utenti devono avere molte password, idealmente una diversa per ogni servizio che utilizza.

> 💡 Come vengono salvate le password in questi file delle password?
> Tipicamente abbiamo tre sezioni:
> 1. **User ID** → ovverosia l'identificatore dell'utente, che deve andare a matchare con la password;
> 2. **SID** → ovverosia un identificatore, che rappresenta il tipo di utente;
> 3. **Hash della password** → quindi la password non viene memorizzata in chiaro, bensì viene memorizzata cifrata.

Questo lo si può osservare nella seguente immagine:

Sorge a questo punto spontanea la domanda: **Come vengono “indovinate” le password?** Vi sono diverse metodologie:

- **Forza bruta** → con un attacco di forza bruta, gli attaccanti usano provano tutte le combinazioni di nome utente/password possibili, al fine di ottenere l'accesso all'account di un utente. Quindi, l’attacco di forza bruta diventa più complicato tanto più la password è lunga. Quasi tutti gli odierni attacchi di forza bruta alle password sono eseguiti da programmi chiamati bot, che possono eseguire ripetutamente attività specifiche per proprio conto, senza alcun intervento umano. Molti di questi attacchi "password-cracking" vengono eseguiti da una botnet, ossia una rete di bot che comprende centinaia di migliaia di computer infettati da malware, tramite il quale un singolo criminale ne prende il controllo. In particolare, gli attacchi di forza bruta si suddividono in due macro-tipologie, ovvero:
  - **Attacchi offline** → negli attacchi offline, l’attaccante ha accesso al materiale crittografato o ad una password hash e tenta una chiave diversa senza il rischio di essere scoperto e quindi senza alcuna interferenza;
  - **Attacchi online** → negli attacchi online, l’attaccante deve interagire con il sistema target, ovvero con il sistema che vuole attaccare. In questi casi, il sistema può contrastare l’attacco, ad esempio, limitando il numero di tentativi di password.
- **Dictionary Attack** → una versione più evoluta e più intelligente dell’attacco di forza bruta è l’attacco utilizzando i dizionari, ovvero: con l’attacco di forza bruta si provano tutte le possibili combinazioni di password, ma naturalmente l’utente utilizzerà **password brevi**, **contenenti parole comuni** e soprattutto **facili da pronunciare**. Quindi, con gli attacchi con il dizionario si vanno a provare password di senso compiuto e associare all’utente, come ad esempio: nomi, nomi di amici e brand di macchine → si vanno, allora, a provare parole contenute nel dizionario, password popolari e password associate all’utente;

> 💡 Come abbiamo appena detto, gli utenti spesso scelgono password che contengono informazioni personali. come il nome di nome di un famigliare o di un animale, in quanto sono facili da ricordare → questo fa sì che l'attaccante ha maggiori possibilità di successo. Spesso gli utenti pensano di utilizzare password semplici e poi sostituire le "o" con gli 0, le "i" con gli 1 e così via → questo non fornisce una difesa più elevata, poichè anche gli attaccanti conoscono questi trucchi e quindi anche questi trucchi sono stati inseriti nel dizionario.

Le possibili contromisure che si possono adottare sono di impostare password con una lunghezza minima; impostare password che siano un mix tra lettere minuscole, lettere maiuscole, simboli e numero; evitare password ovvio (come 1234); cambiare la password regolarmente (ogni 30, 60, 90 giorni) → queste contromisure, in realtà, sono inutili in quanto ad esempio è stato dimostrato che inseriamo i numero o i simboli alla fine della password o comunque con un certo pattern. Le reali contromisure efficaci sono:

  - bloccare l'account dell'utente dopo diversi tentativi di accesso non riuscito;
  - introdurre ritardo temporali tra tentativi di login consecutivi;
  - monitorare i login per rilevare un utilizzo insolito;
  - notificare all'utente tentativi di login;
  - autenticazione a più fattori;
  - verificare se la password è presente in un elenco di parole comuni.

- **Rainbow Tables** → le rainbow tables hanno come obiettivo quello di velocizzare gli attacchi offline, andando a memorizzare le hash delle parole comuni → di conseguenza, questi attacchi sono più veloci nel cracking delle password rispetto agli attacchi di forza bruta e degli attacchi con il dizionario, ma hanno come svantaggio che occupano molta memoria.
- **Ingegneria sociale** → il modo più semplice per ottenere una password è quello di riceverla direttamente dall'utente. Gli utenti scrivono le loro password su appunti, le condividono con i colleghi e utilizzano le stesse password più volte. Esistono attacchi che tentano di ottenere la password o informazioni sensibili relative alle password direttamente dagli utenti e in questo caso si parla di attacchi di **ingegneria sociale** → l'ingegneria sociale prevede lo studio del

comportamento degli utenti al fine di rubare informazioni dell'utente. Gli attacchi di ingegneria sociale sono diversi, come per esempio:

  - **email di phishing** → in cui gli attaccanti inviano un'email con la richiesta di reimpostare la password;
  - **shoulder-surfing** → l'attaccante raccoglie le password osservando alle spalle dell'utente, mentre quest'ultimo si logga al sistema;
  - **dumpster-diving** → l'attaccante cerca nella spazzatura pezzi di carta o documenti password scritte.

  Le **contromisure** a questi attacchi riguardano la sensibilizzazione e la formazione degli utenti.

- Phishing.

Le password hanno un grande problema, ovverosia esse sono **riutilizzabili** ed in questo senso sono stati introdotti due concetti fondamentali, ovvero:

1. **l'invecchiamento delle password** → cerca di garantire che, nel momento in cui la password venga indovinata, essa non sia più valida;
2. **One-Time Password (OTP)** → un caso particolare di password che invecchiano, sono le OTP, ovverosia password che sono valide solo per una sessione di login o una transazione.

Attraverso questi due meccanismi, gli utenti non sono più soggetti ad attacchi di tipo replay, in quanto l'attaccante che ruba la password non può più utilizzarla, in quanto non è più valida. Analizziamo più approfonditamente questi due meccanismi:

- Invecchiamento delle password → l'invecchiamento della password è il requisito per cui una password deve essere cambiata dopo un certo periodo di tempo o dopo il verificarsi di un evento. Si suppone, che il tempo previsto per indovinare una password sia di 180 giorni. Quindi, cambiando la password con una frequenza superiore a 180 giorni, in teoria si ridurrà la probabilità che un aggressore riesca a indovinare una password ancora in uso. In pratica, l'invecchiamento di per sé garantisce poco, perché il tempo stimato per indovinare una password è una media. Se gli utenti possono scegliere password facili da indovinare, la stima del tempo previsto deve cercare un minimo, non una media. L'invecchiamento delle password, quindi, funziona meglio in combinazione con altri meccanismi. L'implementazione dell'invecchiamento delle password presenta dei problemi:
  - Obbligare gli utenti a cambiare le password;

    - Fornire un avviso della necessità di cambiare la password e un metodo facile da usare per farlo;
    - L'invecchiamento della password è inutile se l'utente può cambiare la password con la stessa cosa e quindi si deve essere in grado di registrare le ultime N password inserite.

> 💡 Un approccio alternativo si basa sul tempo → l'utente deve deve cambiare la password con una diversa da quella attuale e la password non può essere cambiata per un periodo di tempo minimo.

- **One-Time Password (OTP)** → la OTP è una password che non è più valida non appena viene utilizzata. Non è vulnerabile agli attacchi replay, ovvero se un attaccante ottiene una password, questa non può essere riutilizzata. Ha bisogno di un token (qualcosa che l'utente possiede) per generare una password casuale una tantum ed utilizza algoritmi matematici che costruiscono una nuova password da quelle precedenti. **L'utente e i sistemi devono essere sincronizzati**.

Supponiamo che l'utente abbia scelto una password forte e che la memorizzi in maniera in modo sicuro. I sistemi, quindi, devono proteggere la password durante il processo di processo di autenticazione, il quale si suddivide in due fasi:

1. inserimento della password → quindi dobbiamo proteggere il campo in cui la password viene inserita. Per proteggere l'inserimento della password, abbiamo bisogno anche che i tempi di risposta siano costanti, in quanto i sistemi controllano la password un carattere alla volta e al primo errore rifiutano la password e questo può essere sfruttato dall'attacante, in quanto osservando il tempo di risposta del sistema è possibile capire da quanti caratteri è composta la password. Quindi, i tempi di risposta del sistema devono essere costanti anche per le password lunghe.
2. memorizzazione della password nel sistema → quindi dobbiamo proteggere l'area di memoria in cui sono memorizzate le copie locali delle password. Inoltre, dobbiamo proteggere i file di log, in quanto succede molte volte che nella fase di login, non riusciamo ad autenticarsi per errori di battitura e quindi se non proteggiamo i file di log, gli attaccanti potrebbero recuperare la password in chiaro dai file di log e correggere gli errori di battitura.

Proteggere le copie locali delle password

Per verificare una password, il sistema deve essere in grado di confrontare le password inserite con le password reali. L'attaccante può tentare di accedere al file contenente le password di sistema. In alcuni di questi sistemi, questo file contiene una tabella con due colonne: una contenente lo user-id e una contenente le password. É, quindi, importante proteggere questo file e per fare questo dobbiamo:

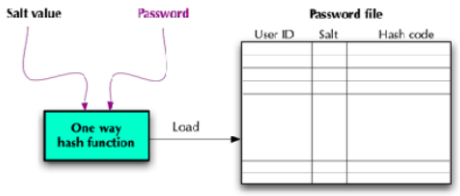
- avere un controllo di accesso limitato → ovverosia solo il SO può accedere al file e di conseguenza non tutti i moduli del SO devono accedere al file delle password;
- consentire l'accesso alla tabella solo ai moduli che ne hanno bisogno;

Da notare che in questo caso non è consigliato fare il backup del file delle password, in quanto il backup potrebbe essere sfruttato dagli attaccanti.

Inoltre, al fine di proteggere la lista delle password da eventuali intrusi, possiamo criptare la lista ed in particolar modo esistono due macro-tipologie di crittografia:

1. **Crittografia standard** → attraverso questa tipologia di crittografia, tutte le tabelle sono crittografate oppure solamente la colonna delle password. Quando l'utente inserisce la password, il sistema decodifica la password criptata e poi le confronta → **la password**, quindi, **è presente in chiaro nella memoria per un breve periodo di tempo**;
2. **Crittografia a senso unico** (One-way cryptography) → attraverso questa tipologia di crittografia, le password nella tabella sono criptate e memorizzate. Quando l'utente inserisce la password nel sistema, quest'ultimo la cripta e poi la compara con quella presente nella tabella. In questo caso, vi è da accertarsi che non sia possibile avere due password con la stessa codifica.

Un'altra cosa che si utilizza è il **Salt value**, il quale cerca di nascondere, all'interno della tabelle delle password, le password ripetute → questo lo si fa, perchè se attraverso un offline attack vedessimo che molti utenti hanno la stessa hash e di conseguenza molti utenti hanno la stessa password, allora gli attaccanti potrebbero concentrarsi sull'indovinare quella password, in quanto indovinando quest'ultima comprometterebbe un maggior numero di utenti. Il Salt value, allora, sono dei bit (tipicamente 2) che vengono aggiunti alla password inserita dall'utente (da notare che il Salt value è associato allo user id e di conseguenza utenti diversi hanno salt value diversi) e in questo modo la funzione di hash viene calcolata sulla password più il salt value → questo comporta, che password uguali avranno salt value diversi e conseguentemente password uguali avranno hash diverse.



Altri mezzi di autenticazione

La determinazione dell'identità, solitamente è basata sulla combinazione di:

- qualcosa che l'utente **conosce** (come ad esempio una password);
- qualcosa che l'utente **ha** (come ad esempio smart card);
- qualcosa che l'utente **è** (come ad esempio l'impronta digitale) → in questo caso si parla di **autenticazione biometrica** ed in particolare ne analizziamo alcune caratteristiche.

I requisiti fondamentali dell'autenticazione biometrica sono quattro:

1. **Universalità** → ovverosia che quasi ogni persona deve avere questa caratteristica;
2. **Distintività** → ovverosia che ogni persona dovrebbe presentare notevoli differenze nella caratteristica;
3. **Permanenza** → ovverosia che la caratteristica non deve cambiare significativamente nel tempo;
4. **Collezionabilità** → ovverosia che la caratteristica deve avere la capacità di essere efficacemente determinata e quantificata.

Le operazioni di una autenticazione biometrica sono:

- **Iscrizione/registrazione (Enrollment)** → ogni persona che deve essere inclusa nel database degli utenti autorizzati, deve prima essere iscritto/registrato al sistema (è l'analogo dell'assegnazione di una password ad un utente). L'utente presenta il nome e tipicamente un qualche tipo di password o PIN e il sistema rileva alcune caratteristiche biometriche dell'utente (come per esempio l'impronta digitale). A questo punto, il sistema digitalizza l'input e poi estrae un insieme di caratteristiche che possono essere memorizzate come un numero o un insieme di numeri, che rappresenta le caratteristiche biometriche uniche dell'individuo. A questo punto l'utente è iscritto al sistema, il quale quindi mantiene per ogni utente:
    - lo user ID;

    - una password o un PIN;
    - il valore biometrico.

Un’autenticazione biometrica tenta di autenticare un individuo in base alle sue caratteristiche fisiche uniche. In genere, i sistemi biometrici incorporano un qualche tipo di sensore o scanner per leggere le informazioni biometriche e poi confrontarle. Le operazioni successive alla registrazione per l’autenticazione biometrica sono:

- La **verifica** → per la verifica biometrica, abbiamo che l’utente inserisce un PIN e utilizza anche un sensore biometrico. Il sensore estrae la caratteristica corrispondente e la confronta con il modello memorizzato per l’utente. Se vi è corrispondenza, allora, il sistema autentica l’utente;
- L’**identificazione** → l’individuo utilizza il sensore biometrico, ma non presenta alcuna informazione aggiuntiva. Il sistema, quindi, confronta il modello inviato con l’insieme dei modelli memorizzati e se vi è corrispondenza, l’utente viene identificato, altrimenti viene rifiutato.

> 💡 Data la complessità delle caratteristiche fisiche non è non è possibile avere una perfetta corrispondenza tra ciò che viene memorizzato e ciò che viene presentato per l'autenticazione.

Infine indichiamo quelle che sono le limitazioni dell’autenticazione biometrica:

- la bassa accuratezza dell’algoritmo di matching → esso può portare ad avere sia falsi positivi (quindi consentire l’accesso a un utente non autorizzato) sia falsi negativi (quindi non consentire l’accesso ad un utente autorizzato);
- falsa falsificazione dei tratti biometrici → in quanto l’impronta digitale viene lasciate in molti luoghi;
- bassa accettazione da parte dell’utente → l’utente potrebbe non gradire, ad esempio, la scansione della retina;
- alti costi.

I vantaggi, invece, sono:

- facili da usare;
- hanno un’ottima user experience;
- non trasferibili.