L’autenticazione che si evolve: dalla password ai token U2F

È ben chiaro a tutti che **le prime misure di sicurezza di un sistema informatico**, grande o piccolo che sia, sono **l’identificazione e autenticazione dell’utente** che si collega per utilizzarlo. Solo una volta identificato ed autenticato, è possibile autorizzare un utente ad utilizzare il sistema.

Sin dai tempi dei primi sistemi informatici si capì come il solo utilizzo di un identificativo dell’utente, oggi chiamato “nome utenza” o “username”, non fosse sufficiente. Infatti il nome utenza permette di identificare, appunto, l’utenza sul sistema ma non di garantire che chi si collega abbia diritto ad utilizzarla. **Il nome utenza non è un’informazione riservata**, anche se non sempre è pubblica. Spesso è il nome della persona; oggi è di frequente l’indirizzo email o un’informazione simile. Per autenticare una persona (ma anche un altro sistema) è necessario utilizzare un segreto che sia noto solo a quella persona (o altro sistema). **Ecco la nascita della “password”**, informazione segreta nota solo alla persona (o sistema) che, in aggiunta allo “username”, permette di identificare ed autenticare chi vuole connettersi ad un sistema informatico.

Password: uso e sicurezza

La sicurezza dell’autenticazione tramite password si basa su alcuni assunti: il principale è che la password sia nota solo a chi deve connettersi e, almeno in parte (vedi di seguito), al sistema cui si vuole accedere. Lo scenario ideale è quello di una persona che tiene a mente la password, non la comunica a nessuno né la scrive su alcun supporto, sia cartaceo che informatico. Quindi **una password deve essere:**

- facile da ricordare
- difficile da indovinare.

Ogni sistema informatico deve archiviare delle informazioni che gli permettono di verificare la correttezza delle password. Sui sistemi le password non devono essere archiviate in chiaro (purtroppo è ancora frequente che questo venga fatto per "semplicità"), ma **devono essere adottate specifiche routine crittografiche** (hash crittografici) **uni-direzionali, ovvero non-reversibili**, che trasformano le password in stringhe all'apparenza pseudo-casuali. Visto che l'algoritmo non è invertibile, se si viene a conoscenza della stringa pseudo-casuale non è possibile risalire alla password applicando una procedura automatica. L'unica possibilità è di provare a indovinare la password e riapplicare l'algoritmo per vedere se la stringa appena generata corrisponde a quella della password che si vuole scoprire. Quindi, se le password sono difficili da indovinare, **è praticamente impossibile risalire dalla stringa alla password**, mentre – data la password in chiaro – è facile verificare che la sua trasformata corrisponde alla stringa memorizzata. In questo modo, **anche in caso di un Data Breach sul sistema informatico e divulgazione delle stringhe, non dovrebbero esserci problemi di sicurezza** in quanto non dovrebbe essere possibile risalire dalle stringhe alle associate password.

Come purtroppo ben sappiamo, questo modello teorico in pratica non funziona per molteplici ragioni. La prima è che **non siamo molto bravi a creare password difficili** da indovinare, anzi dalle periodiche statistiche risulta che sono sempre molto utilizzate password quali "12345", "qwerty", password=username, eccetera. Questo anche perché c'è una chiara dicotomia tra "facile da ricordare" e "difficile da indovinare": **normalmente ciò che è facile da ricordare è anche facile da indovinare!**

A questo è necessario aggiungere il problema pratico di gestire un alto numero di credenziali: 40 anni fa chi

utilizzava sistemi informatici doveva ricordarsi, al più, tre o quattro credenziali. **Oggi ciascuno di noi deve ricordare decine di credenziali**, utilizzate in ambito sia personale che professionale. Una facile soluzione è utilizzare la stessa password per tutte le utenze, ma questo comporta il grave rischio che, una volta carpita la password di un'utenza, un attaccante possa accedere a tutte le utenze utilizzate dalla stessa persona, sia in ambito lavorativo che privato.

Per combattere la tendenza ad utilizzare la stessa password in molteplici utenze, **è comune politica forzare la modifica periodica della password e richiederne una certa complessità**, ovvero lunghezza minima e presenza di lettere, numeri, caratteri di punteggiatura ecc. (si veda ad esempio [1]).

Allo stesso tempo **i programmi di Password Cracking**, basati per lo più sul concetto di provare le password più facili da indovinare e le loro variazioni, diventano sempre più sofisticati, veloci e di facile uso, e gli incidenti di Data Breach – con la diffusione di milioni di credenziali – sono ormai notizie fin troppo frequenti [2].

La situazione è peggiorata a tal punto che, per un utilizzatore, qualunque la gestione delle password è praticamente l'opposto di quello che dovrebbe essere; le password spesso risultano:

- difficili da ricordare
- facili da indovinare da parte di un attaccante!

Tutto questo ha portato anche il NIST nel 2017 a rivedere le proprie linee guida sulla gestione delle credenziali, rimuovendo la richiesta di cambio password periodico e aggiornando il requisito di complessità (si veda [3] per i dettagli).

Gestire le password

Per migliorare la gestione delle password si è compreso, ormai da tempo, che bisogna procedere come minimo in **due direzioni:**

1. ridurre il numero di credenziali che ogni persona deve gestire
2. rendere più semplice la gestione delle password.

In teoria, in ambito aziendale la soluzione esiste. Si tratta di:

a) integrare tutte le applicazioni aziendali in un unico Dominio di Autenticazione aziendale (ad esempio gestito con un LDAP server, Microsoft Active Directory ecc.);

b) estendere il Single-Sign-On (SSO) a tutte (o quasi) le applicazioni;

c) aggiungere, per utenze amministrative o per l'accesso ad applicazioni con requisiti particolari di sicurezza, un ulteriore fattore di autenticazione (discuteremo nel prossimo articolo di Multi Factor Authentication – MFA/2FA).

In questo modo ogni utente aziendale avrebbe **una sola utenza, con una sola password, per accedere a qualunque dispositivo e applicazione aziendale** (con alcune eccezioni quali il PIN della SIM dello smartphone o della carta di credito/bancomat). Questo porterebbe tanti benefici – ovviamente una password per persona, anche se complessa, è molto più facile da ricordare – anche per l'azienda, in quanto **una gestione centralizzata delle credenziali migliora l'efficienza e la sicurezza**, e le applicazioni o sistemi in SSO non devono gestire direttamente le password ed il processo di autenticazione. Il principale rischio è quello insito in ogni gestione centralizzata con la presenza di un "Single Point of Failure": l'accesso a qualunque sistema richiede la disponibilità del Dominio di Autenticazione. Ma la preoccupazione maggiore viene dalla possibilità di un accesso fraudolento: in questo caso

l’attaccante, avendo carpito una credenziale, potrebbe aver accesso con questa identità a tutte le applicazioni e sistemi aziendali (a meno della presenza di ulteriori livelli di autenticazione, MFA ecc.). Data la numerosità dei Data Breach e il rischio di diffusione di credenziali che possano permettere l’accesso alla rete aziendale, **è particolarmente importante che gli accessi da remoto o in mobilità siano soggetti a misure di sicurezza superiori** all’uso della sola password, come vedremo più avanti.

La gestione delle **password personali**, includendo anche quelle che una persona usa per i sistemi informatici aziendali, è sicuramente **più complessa**. Da anni sono presenti sul mercato **applicazioni per la gestione delle credenziali**, i Borsellini delle Password o Password Manager, anche online. Questi strumenti sono molto utili e fondamentalmente permettono di gestire un grande numero di credenziali ricordandosene solo una, quella per accedere al Password Manager stesso. Visto che un utilizzatore di un Password Manager non deve più ricordarsi le password gestite da questo, i Password Manager generano e gestiscono password molto lunghe, pseudo-casuali, con alta complessità, e diverse per ogni account.

I Password Manager sembrerebbero essere gli strumenti che risolvono i problemi di gestione delle password, ma in realtà hanno alcune **notevoli limitazioni** (e, per le versioni online, le sempre possibili vulnerabilità comuni a qualunque servizio web). Il principale problema di utilizzo dei Password Manager è dovuto alla **disponibilità dei dati** e alla **praticità di utilizzo**. Ovviamente non è possibile autenticarsi ad un sistema se non si ha accesso prima al proprio Password Manager e, vista la complessità dei dati, è sempre necessario copiare e incollare (o trascrivere quando questo non è possibile) le credenziali dal Password Manager al sistema ove ci si vuole autenticare. Personalmente utilizzo un Password Manager per gestire alcune centinaia di credenziali, sia per scopi privati che di lavoro, e mi rendo conto quotidianamente di queste

limitazioni pratiche che lo rendono sì uno strumento utile, ma non per tutti.

Per quanto riguarda i **servizi online**, in Cloud ecc., lo sviluppo degli ultimi anni è stato **ridurre il numero di autenticazioni** richieste per l'accesso utilizzando l'equivalente di un SSO aziendale, ovvero la Federazione tra organizzazioni (o SSO Federato). Più precisamente, il **Federated Identity Management (FIM)** permette ad una organizzazione – l'Identity Provider – di registrare, identificare ed autenticare gli utenti facendo in modo che tale autenticazione sia riconosciuta valida anche da altre organizzazioni. L'Identity Provider garantisce ad altre organizzazioni l'identità e la validità dell'autenticazione dell'utente in modo che non sia necessaria un'ulteriore registrazione ed autenticazione. Questo è **quello che succede quando si accede ad un servizio web utilizzando il bottone "Autenticati con …"** (ad esempio Facebook o Google). Sommariamente, l'utente prima si autentica con l'Identity Provider; poi può accedere, senza autenticarsi di nuovo e senza ulteriori credenziali o password, a servizi web Federati a questo Identity Provider. Infatti l'Identity Provider prova al servizio web Federato che l'autenticazione è valida inviando token crittografici riconosciuti. Vi sono **alcuni protocolli crittografici** che permettono di fare questo, tra cui SAML, OpenID, Oauth, WS-Federation ecc. Ad esempio, in Italia, il Sistema Pubblico di Identità Digitale (SPID) si basa sul protocollo SAMLv2. Si noti come l'accesso Federato copra tipicamente le fasi di identificazione e autenticazione dell'utente, mentre l'autorizzazione rimane in carico ad ogni organizzazione ed applicazione.

La Federazione tra organizzazioni permette quindi di uniformare e semplificare gli accessi ai servizi web, riducendo il numero di credenziali che l'utente finale deve gestire.

Va però tenuto conto anche di un altro aspetto importante, che

riguarda la **privacy degli utenti:** un Identity Provider traccia gli accessi dei propri utenti a tutte le organizzazioni e applicazioni Federate, ed è quindi importante che queste informazioni, che potrebbero portare ad un monitoraggio delle attività delle persone, siano gestite nel pieno rispetto della privacy.

Come superare le password

Ovviamente la ricerca di diverse modalità di autenticazione è vecchia quasi quanto le password stesse. È ben noto che, in generale, il processo di autenticazione si può basare su **tre tipi distinti di informazioni** segrete o uniche:

1. qualcosa che si sa: ad esempio la password;
2. qualcosa che si è: una caratteristica biometrica;
3. qualcosa che si ha: una chiave o altro oggetto caratteristico.

L'utilizzo di tutti e tre i tipi di informazioni per l'autenticazione è ben noto nella vita di tutti i giorni. Sino a poco tempo fa, nei sistemi informatici l'autenticazione biometrica e con una chiave fisica sono state ristrette ad applicazioni ed esigenze particolari.

Biometria

L'autenticazione biometrica [4] si basa sul riconoscimento di una caratteristica unica del corpo umano, quale ad esempio **l'impronta digitale, la forma del volto, la voce, l'iride, la geometria della mano** eccetera. L'autenticazione biometrica richiede una fase iniziale in cui viene **registrato un campione** della caratteristica biometrica, detto template, ed **associato all'identità da autenticare.** Il template è archiviato sul sistema che esegue l'autenticazione. Quando poi l'utente vuole accedere al sistema, viene rilevato un altro campione della caratteristica biometrica dell'utente che viene confrontato con il template registrato per quell'utente (processo di

“verifica biometrica”)[5]. Se il confronto è positivo, ovvero il campione ed il template sono sufficientemente simili, l’autenticazione biometrica è riuscita.

L’autenticazione biometrica comporta **molte problematiche** che ne hanno reso difficile l’adozione diffusa. Le seguenti sono alcune delle principali.

Mentre la verifica della password è svolta con una procedura che non ha errori od incertezze – o la password inserita è identica a quella registrata o è diversa – nel caso di autenticazione biometrica **due campioni della stessa caratteristica non sono mai identici.** Ogni sistema biometrico deve quindi definire una soglia di somiglianza/errore tra il campione ed il template. Questo però porta a **due tipi di errori:**

- i Falsi Positivi, ovvero un campione dichiarato simile al template ma appartenente ad un’altra persona (False Match Rate, FMR)
- i Falsi Negativi, ovvero un campione dichiarato diverso dal template ma appartenente alla stessa persona (False Non-Match Rate, FNMR).

Alzando la soglia di somiglianza/errore, i Falsi Positivi diminuiscono ma al contempo aumentano i Falsi Negativi, ovvero chi non riesce ad autenticarsi pur avendone diritto. Abbassando la soglia di somiglianza/errore avviene il contrario, si veda Fig. 1.

Fig 1: Diagramma di FMR e FNMR al variare della soglia di somiglianza/errore (tratta da [4])

Inoltre, i rilevatori biometrici possono essere soggetti a **vulnerabilità che possono portare a Falsi Positivi**, come ad esempio la ricostruzione di impronte digitali con gelatina o la ricostruzione di volti a partire da immagini ad alta

risoluzione, che sono state riconosciute come corrette verifiche in noti esperimenti.

Ma le principali problematiche della biometria risiedono sui **rischi della gestione dei template e delle informazioni biometriche.** Infatti una caratteristica biometrica non può essere modificata o sostituita, come invece facciamo regolarmente con le password. Pertanto, **un Data Breach di template biometrici potrebbe avere conseguenze molto gravi per la privacy degli utenti.** Quindi un sistema informatico che gestisce centralmente l'autenticazione biometrica richiede misure di sicurezza molto avanzate [6]. È quindi **meno rischioso archiviare i template biometrici localmente**, presso l'utente stesso, come per esempio nel passaporto o nei più recenti smartphone.

Infine vi sono stati anche **casi estremi** ove, ad esempio, pur di riuscire a rubare un'automobile protetta con autenticazione biometrica ad impronte digitali, i ladri hanno amputato un dito al proprietario [7].

Sistemi a chiave pubblica+privata

La crittografia a chiave pubblica+privata offre **un'altra possibile alternativa alla password**. L'idea è che l'utente è in possesso di una chiave privata, che deve mantenere segreta, alla quale è associata una chiave pubblica. Le due chiavi sono legate univocamente una all'altra da un algoritmo crittografico ma dalla chiave privata è possibile ottenere quella pubblica, mentre il contrario non è possibile: ovvero **non è possibile ottenere la chiave privata conoscendo quella pubblica**. Inoltre quanto cifrato con una chiave (pubblica/privata) può essere decifrato solo con l'altra chiave (privata/pubblica). Sui sistemi a cui l'utente vuole connettersi, la chiave pubblica viene associata allo username intestato all'utente.

L'idea del meccanismo di autenticazione è **molto semplice**,

anche se i protocolli crittografici reali sono più complessi a garanzia della sicurezza: quando l'utente si collega al sistema con il proprio identificativo (username), il sistema gli invia una stringa pseudo-casuale unica. L'utente cifra con la propria chiave privata la stringa e la invia, cifrata, al sistema, che decifra con la chiave pubblica i dati ricevuti e verifica che la stringa ottenuta sia la stessa che ha inviato. In questo modo l'utente dimostra al sistema di essere in possesso della chiave privata, dato che solo quanto cifrato dalla chiave privata può essere decifrato dalla corrispondente chiave pubblica.

Mentre **l'autenticazione con password è la condivisione di un segreto** in quanto la password deve essere inviata ed archiviata dal sistema, anche se protetta come hash, **il sistema di autenticazione a chiave pubblica+privata non invia o archivia nulla di segreto sul sistema**. Pertanto, i rischi di Data Breach sono molto inferiori.

L'autenticazione a chiave pubblica+privata ha però alcuni **svantaggi notevoli** rispetto alla password. L'utente deve mantenere segreta la chiave privata, questa però è una stringa molto lunga (sino anche a mille caratteri), pseudo-casuale e con particolari strutture matematiche che dipendono dall'algoritmo crittografico utilizzato. Ovviamente non è possibile mantenere a memoria le chiavi private, che sono sicuramente **impossibili da indovinare ma anche impossibili da ricordare.**

Il problema, quindi, si sposta interamente su come l'utente possa mantenere segreta e personale la chiave privata. L'idea di protocollo descritta precedentemente prova solamente che l'utente che si autentica è in possesso della chiave privata. L'utente deve, perciò, mettere in pratica delle misure di sicurezza che garantiscano che nessuno possa copiargli la chiave privata.

La soluzione per la gestione sicura della chiave privata è la

smart-card (o chip-card, o integrated-circuit-card ICC). Ben conosciamo questi oggetti in quanto costituiscono ormai la base della maggior parte del commercio, ovvero le carte di credito, debito, bancomat ecc. con “chip”, anche contactless, e le SIM telefoniche.

La smart-card è fondamentalmente un **circuito integrato con particolari caratteristiche di sicurezza**: al suo interno è archiviata una chiave privata che non può essere estratta né letta, neanche con attacchi fisici. Il processore incluso nella smart-card permette la firma o cifratura di dati in input con la chiave privata archiviata in esso. Fondamentalmente, una smart-card attesta che al suo interno è presente una particolare chiave privata, e garantisce che nessuno possa copiarla dal suo contenitore.

Le misure di sicurezza che un utente deve mettere in pratica per la chiave privata archiviata in una smart-card devono **garantire che la smart-card non sia persa o rubata.** Queste misure sono le stesse che bisogna adottare per proteggere una chiave tradizionale: infatti una smart-card implementa il **terzo tipo di informazione** per l’autenticazione, basato su **“qualcosa che si ha”.**

Come puro strumento di autenticazione la smart-card ha avuto **un’adozione e una diffusione limitate**. Vi sono esempi di adozione anche su larga scala, ad esempio il Servizio Sanitario Nazionale e la Tessera Sanitaria/Carta Regionale dei Servizi personale. L’utilizzo della smart-card è però **limitato a servizi che richiedono un alto livello di sicurezza**, come appunto quelli sanitari, ove il costo della carta, del lettore da utilizzare e del software da installare su apposti computer può essere sostenuto in vista dei rischi a cui si possono esporre i dati.

Tenuto conto del rischio di furto o smarrimento di una smart-card, per servizi che richiedono un alto livello di sicurezza è necessario associare l’autenticazione con smart-card con un

altro tipo di autenticazione, ad esempio la tradizionale password, realizzando un **sistema di autenticazione a multi-fattori (MFA)**. Infatti, per utilizzare sia le carte di credito/debito che le SIM telefoniche è necessario l'inserimento di un codice segreto di qualche cifra (PIN) tenuto a memoria come una password. Questo serve a garantire che l'utilizzatore della smart-card sia proprio l'utente/legittimo proprietario e a concludere la catena di autenticazione dalla persona al sistema informatico.

Nel prossimo articolo descriveremo i più recenti sviluppi nei metodi di autenticazione, anche per cercare di capire **se si stia avvicinando il momento in cui potremo almeno ridurre le problematiche legate alla gestione delle password.**

Riferimenti Bibliografici

[1] D.lgs. 30 giugno 2003, n. 196, Allegato B.

[2] Per un recente esempio si veda Wired, *"Hackers are passing around a megaleak of 2.2 billion records"*, 2019/01/30, https://www.wired.com/story/collection-leak-usernames-passwords-billions/.

[3] NIST Special Publication 800-63B *"Digital Identity Guidelines: Authentication and Lifecycle Management"*, giugno 2017, in particolare la sezione 10.2.1, https://pages.nist.gov/800-63-3/sp800-63b.html.

[4] Per una introduzione alla biometria si veda A.K. Jain, A. Ross, S. Prabhakar "An Introduction to Biometric Recognition", IEEE Transactions on circuits and systems for video technology, Vol. 14, n. 1, Jan. 2004, https://www.cse.msu.edu/~rossarun/pubs/RossBioIntro_CSVT2004.pdf.

[5] Invece per la "identificazione biometrica" viene fornito solo il campione biometrico e tramite una ricerca sui template disponibili si vuole scoprire l'identità dell'utente a cui il

campione appartiene.

[6] Garante per la Protezione dei Dati Personali, *"Provvedimento generale prescrittivo in tema di biometria"*, 12 novembre 2014, Pubblicato sulla Gazzetta Ufficiale n. 280 del 2 dicembre 2014, https://www.garanteprivacy.it/web/guest/home/docweb/-/docweb-display/docweb/3556992.

[7] BBC News "Malaysia car thieves steal finger", 31 marzo 2005, http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm.

Articolo a cura di **Andrea Pasquinucci**

https://www.ictsecuritymagazine.com/articoli/lautenticazione-che-si-evolve-dalla-password-ai-token-u2f/