L’autenticazione che si evolve: dalla password ai token U2F – Parte 2

Nell’articolo precedente sono state brevemente riassunte le principali caratteristiche del processo di autenticazione tramite password, biometria e chiavi pubbliche+private. La problematica principale è però non tanto definire delle tecniche di autenticazione teoricamente sicure, quanto trovare dei processi di autenticazione che siano facilmente adottabili e che garantiscano un livello di sicurezza adeguato ai rischi dei relativi servizi informatici.

Autenticazione a Fattori Multipli (MFA)

La possibilità di utilizzare tipi di autenticazione diversi (qualcosa che *si ha, si sa, si è*), unita alle criticità dell’utilizzo della password e della necessità per alcuni servizi di aumentare il livello di sicurezza del processo, porta a considerare l’utilizzo di più tipi di autenticazione per la stessa richiesta di accesso. Nella pratica si presentano principalmente due tipologie di MFA:

1. vengono richiesti in serie due (molto raramente tre) fattori di autenticazione al momento del primo accesso al servizio o sistema: ad esempio prima una richiesta di username+password e, se questa ha successo, la richiesta di autenticazione biometrica o di autenticazione tramite un token che si ha, ad esempio, tramite una smartcard; se entrambe le autenticazioni hanno successo, viene stabilita una sessione e si ha accesso senza restrizioni al servizio;
2. al momento del primo accesso al servizio o sistema viene richiesto un fattore di autenticazione, ad esempio username con password o autenticazione biometrica; se

l’autenticazione ha successo, viene stabilita una sessione e si ha accesso al servizio ma limitatamente a funzioni con un livello di rischio basso, ad esempio solo in lettura; per poter eseguire delle attività a maggior rischio (scrittura, modifica dei dati o attività dispositive) viene richiesta un’ulteriore autenticazione al momento stesso dell’attività e per ogni esecuzione di attività a rischio elevato.

Ovviamente queste sono solo due tipologie di base di MFA e, in pratica, sono spesso adottate molteplici varianti.

Un concetto importante da sottolineare è la necessità per attività che comportano un livello di rischio non basso, di autenticare la singola transazione e non basarsi unicamente sull’autenticazione dell’intera sessione. Per capire la necessità di autenticare la singola transazione è necessario descrivere brevemente alcuni tipi di attacchi.

Attacchi di Session Hijacking, Replay, Man in the Middle (MitM) e Phishing

Quando ci si autentica presso un servizio viene usualmente creata una sessione, ovvero all’utente vengono assegnati dei codici temporanei che lo identificano per la durata della sessione in modo da non dover ripetere il login ad ogni transazione. Nel caso di navigazione Web questo è tipicamente realizzato con dei Cookie (SessionID) contenenti del materiale crittografico creato dal Server, inviati e archiviati temporaneamente nel Browser. Ad ogni richiesta al Server, il Browser allega i Cookie di sessione in modo che il Server identifichi e autentichi la richiesta senza dover ripetere il processo di login. Ma diversi tipi di Malware sul dispositivo dell’utente possono intercettare i Cookie e, in assenza di altre misure di sicurezza, utilizzare i dati di sessione per accedere al servizio al posto dell’utente, in pratica una sottrazione e appropriazione della sessione utente

(**“hijacking”**).

Un altro tipo di attacco, ormai raramente efficace, consiste nell’intercettare i dati o il traffico di autenticazione dell’utente, anche se cifrato, e riutilizzarlo (attacco **“replay”**). Per evitare questo tipo di attacco tipicamente si allega del materiale unico per ogni autenticazione, che non può essere riutilizzato.

Per evitare questi tipi di attacchi bisogna quindi non solo autenticare con materiale unico ogni sessione, ma anche autenticare indipendentemente – meglio se con un diverso tipo di autenticazione – ogni singola transazione con livello di rischio non trascurabile. L’autenticazione di una sessione deve permettere l’accesso per un periodo di tempo esteso (ma non illimitato) che può andare dai pochi minuti a giorni o anche mesi a secondo del livello di rischio dell’applicazione, mentre l’autenticazione di una singola transazione deve essere molto limitata nel tempo e non riutilizzabile.

Infine, l’attacco più pericoloso e difficile da sconfiggere consiste nell’impersonare il servizio (**“Phishing”**) o riuscire a intercettare la connessione al servizio inserendosi al suo interno (**“Man in the Middle”**)[1]. In entrambi i casi, l’effetto è che l’attaccante vede passare (in chiaro) le informazioni di autenticazione, ovvero lo username+password, i cookie e qualunque altro dato, e può utilizzarli a suo piacimento.

Autenticazione e codici One Time (OTP)

L’evoluzione degli attacchi e delle contromisure di sicurezza ha portato a una specie di gara di velocità (“race condition”). Visto che username+password sono a relativamente alto rischio di essere intercettate e riutilizzate anche a distanza di giorni, mesi o anni, per transazioni a rischio non basso è necessario autenticare ogni singola transazione con un codice valido solo per quella transazione (OTP) e per un

limitato periodo di tempo. Il primo metodo adottato a questo scopo è quello delle chiavette (Token) che generano codici pseudo-casuali, ormai abbastanza comuni per gli accessi ai servizi online bancari, anche se recentemente in via di dismissione.

L'idea di base è abbastanza semplice: il Token mantiene in hardware un numero segreto ("seed") a partire dal quale un algoritmo crittografico, tipicamente di Hash, periodicamente (ad esempio ogni 30 o 60 secondi) genera un numero pseudo-casuale di cui mostra all'utente una parte (usualmente 6 cifre). Un'elaborazione parallela viene eseguita dall'Hardware collegato al Server che eroga il servizio di accesso. Quando è necessario autenticare una transazione, l'applicazione richiede all'utente l'inserimento del codice prodotto in quel momento dal Token, che viene verificato con il corrispondente codice prodotto sul Server. Il Token fornisce un secondo tipo di autenticazione ("qualche cosa che si ha") rispetto alla password ("qualche cosa che si sa"), è valido solo per quella transazione e la sua validità è limitata nel tempo.

Questa soluzione sembrerebbe a prima vista garantire ottimi livelli di sicurezza. Purtroppo presenta due debolezze: i costi dell'Hardware e di gestione, e una "race condition". È facile immaginare i costi di questa soluzione pensando a un servizio, ad esempio bancario, con milioni di clienti disseminati geograficamente: anche se il costo di ogni Token fosse di pochi euro, con milioni di clienti il solo costo dei Token ammonterebbe a milioni di euro, a cui aggiungere i sistemi server e la gestione del supporto ai clienti. Non è detto, quindi, che questa sia la soluzione di sicurezza più efficace dal punto di vista della gestione dei rischi e dei costi/benefici.

Un'alternativa meno costosa è quella di inviare un codice OTP a tempo al telefono cellulare dell'utente via SMS. In questo caso è il telefono cellulare che ha il ruolo di "qualche cosa che si ha", anche se il codice è generato sul server e inviato

al telefono su un canale diverso da quello utilizzato per l'accesso al servizio. Ovviamente molto meno costosa, questa soluzione è anche meno sicura[2] in quanto il protocollo SMS non garantisce né la consegna del messaggio né la sua confidenzialità. Infatti vi sono stati casi di attacchi alle infrastrutture telefoniche per intercettare gli SMS con codici di autenticazione OTP [1]. Come sempre, il rapporto costi/benefici di sicurezza è l'argomento principale per scegliere una qualunque soluzione.

Abbiamo visto che il codice OTP può essere utilizzato una volta sola e dura un limitato periodo di tempo, ma gli attaccanti, a costo di spese maggiori, possono aggirare questa difficoltà: è sufficiente fare hijacking della sessione ed intercettare il codice OTP inviato dall'utente utilizzandolo nella propria sessione. Il tutto richiede tempi ben precisi, per transazioni su siti di banche online nell'ordine della decina di secondi, ma possibili.

Si noti come per l'OTP via SMS un punto di debolezza sia già l'invio del codice al telefono cellulare, mentre per la soluzione a Token la debolezza è nella possibile intercettazione del codice o direttamente sul Browser dell'utente oppure tramite siti di Phishing [2,3].

La rinascita della biometria

Negli ultimi anni l'evoluzione dei processi di autenticazione ha avuto principalmente due obiettivi:

1. semplificare i processi di autenticazione per gli utenti
2. ridurre i costi.

È sicuramente un problema per tutti ricordare una gran quantità di codici, PIN, username+password eccetera. Siamo tutti invece molto soddisfatti dell'utilizzo delle smartcard contactless, che ci permettono di fare acquisti per piccole cifre avvicinando la carta di credito al lettore, e

null’altro. In questo caso ci si autentica unicamente con “qualche cosa che si ha” e il rischio principale è la perdita o furto della carta di credito. Nel caso di furto, questo particolare rischio economico per l’utente e per l’emittente della carta di credito è però basso e ampiamente compensato dall’aumento delle transazioni[3].

Al contempo, lo sviluppo negli ultimi anni dei lettori di caratteristiche biometriche ha permesso la produzione di dispositivi a basso costo e relativa buona efficienza che sono stati integrati nei prodotti per i consumatori. Ormai non solo i PC portatili di alta gamma ma anche moltissimi smartphone sono dotati di un lettore delle impronte digitali o delle caratteristiche del viso. Invece di ricordarci un username+password o un PIN, basta appoggiare il dito al lettore biometrico dello smartphone per aver accesso al telefono e alle App che sfruttano questa funzionalità, prime di tutte le App bancarie ma anche molte altre [5]. Le problematiche di Privacy sono per lo più risolte dal fatto che i dati biometrici sono raccolti e mantenuti da apposite componenti hardware sul dispositivo dell’utente, e non sono gestite da un sistema centrale.

Un possibile approccio è quindi quello di utilizzare username+password come metodo di autenticazione di riserva, che si può mantenere in busta chiusa in caso di difficoltà, ma utilizzare quotidianamente l’autenticazione biometrica per accedere a questi dispositivi e alle relative applicazioni.

Ovviamente non stiamo facendo autenticazione a più fattori, ma sostituendo l’utilizzo di un fattore scomodo con uno più funzionale. È importante notare anche che quanto appena descritto riguarda l’accesso al dispositivo, lo smartphone, non l’accesso a servizi remoti come quelli descritti nella sezione precedente. Accedere ad uno smartphone con l’impronta digitale non risolve direttamente il problema di autenticare le transazioni economiche sul sito della banca.

Autenticazione mutua

Il vero problema dell'autenticazione delle transazioni online risiede nella reale autenticazione reciproca tra client e server: ovvero l'applicazione deve essere sicura di chi sia l'utente che la contatta e viceversa l'utente deve essere sicuro di comunicare direttamente con l'applicazione, senza alcuna interferenza.

Il protocollo SSL/TLS prevede a questo scopo l'utilizzo di certificati client, similmente ai certificati server che ben conosciamo, ma la complessità della loro gestione – ad esempio tramite smartcard – non ne ha favorito l'adozione.

Un recente approccio alternativo è quello di utilizzare lo smartphone come oggetto di seconda autenticazione tramite un App Authenticator, quali quelle di Google, Microsoft, eccetera o della propria banca. L'idea è quella di sostituire il Token o l'SMS descritto precedentemente con lo smartphone sul quale è installata una App di autenticazione. Questa App associa lo specifico smartphone, con il proprio numero di telefono, ad un utente del servizio. Inoltre questa App si autentica mutualmente con il servizio online. Quando l'utente esegue tramite PC una transazione sul servizio online, il servizio online richiede alla specifica App associata all'utente la conferma della transazione. Grazie alla mutua autenticazione, il servizio online è sicuro di comunicare con l'App sullo smartphone dell'utente. L'utente non deve far altro che avere a disposizione lo smartphone e confermare la richiesta di autorizzazione inserendo il PIN di accesso all'App, oppure l'impronta digitale o il riconoscimento facciale. Facendo in questo modo si può anche raggiungere un'autenticazione a tre fattori: username+password per l'accesso al servizio online ("qualcosa che si sa"), lo smartphone ("qualcosa che si ha") e l'autenticazione biometrica ("qualcosa che si è") per accedere allo smartphone.

Le Authenticator App, seppur molto più sicure dell'invio di un

SMS, hanno però anch'esse alcune limitazioni: a parte la possibilità di smarrimento, furto o anche solo scaricamento della batteria dello smartphone, il sistema operativo e le applicazioni per smartphone non sono esenti da vulnerabilità, malware ecc. Ad esempio in caso di rooting, jailbreaking o unlocking, un'App malevola potrebbe interferire con l'esecuzione dell'App di autenticazione portando comunque alla esecuzione di frodi.

Ma forse la principale vulnerabilità degli smartphone è quella di sostituire, in molti casi, il PC e diventare l'unico dispositivo di accesso ai servizi online: in questo caso sia la transazione che tutte le autenticazioni sono svolte sullo stesso dispositivo. Accedendo ad esempio con l'impronta digitale, abbiamo un solo tipo di autenticazione ("qualcosa che si è") eseguita localmente dal dispositivo e non dal servizio online, che deve basarsi solo sull'associazione del dispositivo all'utente fatta al momento di installazione dell'App. In caso di furto, cloning o presenza di malware a questo punto è teoricamente sempre possibile interferire con il processo di autenticazione.

Ovviamente l'adozione di una Authenticator App ha dei costi, in particolare per lo sviluppo e la gestione del software, dei vantaggi, svantaggi e rischi sia per il fornitore di servizi che per gli utenti. È comunque da notare che l'utilizzo dello smartphone dell'utente, a la BYOD, rende il fornitore di servizi dipendente da qualche cosa – lo smartphone scelto dall'utente – che non può gestire né realmente controllare, introducendo un ulteriore fattore di rischio.

Security Keys e FIDO2/U2F

La FIDO Alliance ha proposto un approccio all'autenticazione sicura online alternativo o complementare al Single Sign On (SSO) Federato gestito da un Identity Provider (come discusso nell'articolo precedente), puntando alla realizzazione di un sistema diffuso, non centralizzato, e garante della Privacy

dell’utente. La soluzione proposta dalla FIDO Alliance [6], che va sotto il nome di FIDO2, comprende un gruppo di standard tra cui U2F (CTAP1), UAF, CTAP2 e W3C WebAuthn, ed è basata su molte delle tecnologie di cui abbiamo trattato, dalle smartcard ai Token fisici, ponendosi come obiettivo l’autenticazione mutua utente-servizio.

Il processo di autenticazione è basato su di una Security Key fisica con delle caratteristiche interne simili a quelle di una smartcard, ma che può comunicare a seconda del modello sia tramite USB, sia NFC (come le carte di credito contactless) che Bluetooth Low Energy (BLE). Lo scopo delle Security Key è molteplice [7,8]: sostituire i Token OTP descritti precedentemente, ridurne i costi di gestione, semplificare e automatizzare la gestione per l’utente, aumentare il livello di sicurezza implementando anche l’autenticazione mutua.

Per l’utente l’utilizzo di una Security Key comporta molti benefici: la stessa Security Key può essere utilizzata per autenticarsi a molti diversi servizi online e in qualunque momento può essere svolta la registrazione di un nuovo servizio; può essere utilizzata indifferentemente con PC desktop, portatili, tablet e smartphone senza necessità di particolari dispositivi di collegamento. Svolge quindi le funzioni di una molteplice chiave fisica per il mondo digitale. Inoltre tipicamente una Security Key presenta un bottone, o un lettore di impronte digitali, e agisce solo se attivata da questo.

Ad alto livello il protocollo FIDO2 è abbastanza semplice [7], le principali funzioni sono due: **registrazione** e **autenticazione**. Lo scenario è quello di un utente che si collega a un servizio online Web ed effettua prima la registrazione poi, successivamente, utilizza il servizio. Consideriamo il caso più semplice: quello di un servizio che utilizza la Security Key come secondo fattore di autenticazione per transazioni a rischio non basso (Universal Second Factor, U2F).

L’utente crea inizialmente la sua utenza con username+password e poi procede ad aggiungere il secondo fattore di autenticazione. Per far questo è in possesso di una Security Key che connette al dispositivo che sta usando per accedere al servizio online. L’utente attiva nel servizio online la registrazione della propria Security Key. Le principali attività che seguono sono queste (Fig. 1):

Fig. 1: Registrazione di una Security Key [7]

- il servizio online invia al Browser (o App) dell’utente una stringa pseudo-casuale (“challenge”) e l’istruzione di registrare la Security Key;
- Il Browser (o App) identifica la Security Key connessa al dispositivo, le invia i dati ricevuti dal servizio online e delle informazioni sul servizio quali il suo indirizzo (“origin”), e richiede la creazione di una nuova chiave pubblica+privata;
- La Security Key, dopo la pressione del bottone, crea una chiave pubblica+privata con un identificativo (“handle”), con la chiave privata firma i dati ricevuti e invia la chiave pubblica e i dati firmati al Browser (o App);
- Il Browser (o App) inoltra i dati ricevuti dalla Security Key al servizio online;
- Il servizio online, verificato con la chiave pubblica ricevuta che la firma effettuata dalla Security Key sui dati (ed in particolare sul challenge) sia corretta, archivia la chiave pubblica con il suo identificativo associandoli all’utenza.

D’ora in avanti il servizio online può utilizzare la chiave pubblica ed il suo identificativo per richiedere una seconda autenticazione all’utente. Questo procede in maniera simile, come segue (Fig. 2):

Fig. 2: Autenticazione tramite Security Key [7]

- il servizio online invia al Browser (o App) dell'utente una stringa pseudo-casuale ("challenge") e l'identificativo ("handle") della chiave pubblica associata all'utente che vuole autenticare;
- Il Browser (o App) identifica la Security Key connessa al dispositivo, le invia i dati ricevuti dal servizio online e delle informazioni sul servizio quali il suo indirizzo ("origin"), e richiede l'autenticazione;
- La Security Key, dopo la pressione del bottone, verifica che l'identificativo corrisponda a una propria chiave, che corrisponda ai dati del servizio per cui l'ha creata, firma i dati ricevuti con la chiave privata e invia i dati firmati al Browser (o App);
- Il Browser (o App) inoltra i dati ricevuti dalla Security Key al servizio online;
- Il servizio online verifica con la chiave pubblica dell'utente che la firma effettuata dalla Security Key sui dati sia corretta, e quindi procede con l'esecuzione della transazione.

Si noti come in questo processo il servizio online si fidi solo della firma della Security Key, ovvero dell'attestazione della presenza della chiave privata nella Security Key. D'altra parte, la Security Key verifica sia i dati inviati direttamente dal servizio online che quelli forniti dal Browser: la Security Key esegue la firma solo se tutti i dati ricevuti sono consistenti. In questa maniera abbiamo due attori che in maniera automatica verificano reciprocamente la propria identità: il servizio online e la Security Key. Questo indipendentemente da chi possa cercare di intercettare o modificare la comunicazione.

Ovviamente questa soluzione non riesce a coprire tutti i possibili scenari di attacco: la fase più delicata è sicuramente quella della registrazione iniziale, ove un

attaccante potrebbe cercare di immedesimare il servizio online. Si noti anche che, quando l'utente autorizza la firma della Security Key premendo il suo bottone (o verificando l'impronta digitale), non ha la possibilità di verificare il contenuto dei dati firmati su di un display sicuro; un attaccante potrebbe quindi cercare di modificare i dati della transazione, ad esempio cambiandone il destinatario, prima della firma.

Rimane da chiarire un aspetto comunque importante: come revocare le chiavi generate da una Security Key in caso, ad esempio, di furto o smarrimento della Security Key stessa, o per rimuovere una chiave generata dalla Security Key ma che non si vuole più utilizzare. Il protocollo al momento non prevede la possibilità di cancellare chiavi dalla Security Key né di notificare la perdita (o il ritiro) di una Security Key. Per garantire la Privacy degli utenti, non vi è un identificativo univoco di ogni singola Security Key che altrimenti permetterebbe di tracciare le attività dell'utente. Lo scenario più critico è sicuramente quello del furto di una Security Key e, al momento, il protocollo lascia al gestore di ogni servizio online decidere come gestire questo evento.

Un'ultima osservazione su questo protocollo: le Security Key permettono di gestire un numero arbitrario di servizi online, si pone pertanto il problema della conservazione delle chiavi private sul piccolo dispositivo poiché potrebbero occupare parecchio spazio all'interno del processore sicuro. Lo standard non impone una soluzione ma offre un approccio: mantenere nel processore sicuro una (o due) master key private ed esportare in maniera sicura le altre chiavi private. In questo caso, tutte le chiavi private delle coppie pubblico+privato generate dalla Security Key sono cifrate e decifrate con la master key privata solo all'interno del processore sicuro e possono essere esportate solo se cifrate. Le chiavi private cifrate possono quindi essere archiviate o in un'area di memoria apposita sulla stessa Security Key, o

inviate al servizio online che le archivia insieme alla chiave pubblica. La sicurezza è fornita dal fatto che solo la master key privata all'interno della Security Key può decifrare ed utilizzare queste chiavi.

L'adozione delle Security Key da parte di Google Inc. [7,8] e di molte altre aziende sta dando risultati soddisfacenti sia per la semplicità d'uso che per i livelli di sicurezza raggiunti. Il fatto di essere una "chiave" che possiamo aggiungere al nostro portachiavi, utilizzabile con molteplici nostri dispositivi e diversi servizi online, sicuramente può semplificarci la vita oltre ad aumentare la sicurezza dell'accesso ai servizi.

È però troppo presto per dichiarare che abbiamo finalmente individuato la strada per abbandonare definitivamente le nostre vecchie e care password.

Note:

[1] Alcuni Malware (ad esempio bancari) si inseriscono tra il frontend del Browser e il sistema operativo sul dispositivo dell'utente per accedere, e anche modificare, i dati della connessione al servizio remoto.

[2] Per il momento non consideriamo il caso di accesso al servizio tramite App sullo stesso smartphone.

[3] Per pagamenti di cifre superiori, sono in corso sperimentazioni di carte di credito contactless con autenticazione biometrica delle impronte digitali, si veda ad esempio [4], invece della richiesta del PIN.

Riferimenti Bibliografici

[1] Per un recente caso si vedano:

[2] B. Schneier, "Hacking Two-Factor Authentication", https://www.schneier.com/blog/archives/2009/0

9/hacking_two-fac.html.

[3] Si vedano, ad esempio:

- Ars Technica, “Iranian phishers bypass 2fa protections offered by Yahoo Mail and Gmail”, https://arstechnica.com/information-technology/2018/12/iranian-phishers-bypass-2fa-protections-offered-by-yahoo-mail-and-gmail/.
- The Hacker News ,“WARNING – New Phishing Attack That Even Most Vigilant Users Could Fall For”, https://thehackernews.com/2019/02/advance-phishing-login-page.html.

[4] “Intesa Sanpaolo and Mastercard introducing the first contactless biometric payment card in Italy”, https://www.world.intesasanpaolo.com/hp-news/intesa-sanpaolo-mastercard-introducing-first-contactless-biometric-payment-card-italy/.

[5] Si veda ad esempio “WhatsApp can now be locked using Face ID or Touch ID”, http://www.theverge.com/2019/2/4/18210197/whatsapp-touch-id-face-id-security.

[6] https://fidoalliance.org/ , https://www.w3.org/TR/webauthn/.

[7] J. Lang et al., “Security Keys: Practical Cryptographic Second Factors for the Modern Web”, Google Inc., http://fc16.ifca.ai/preproceedings/25_Lang.pdf.

[8] Ars Technica, “This low-cost device may be the world’s best hope against account takeovers”, https://arstechnica.com/information-technology/2016/12/this-low-cost-device-may-be-the-worlds-best-hope-against-account-takeovers/.

Articolo a cura di **Andrea Pasquinucci**

https://www.ictsecuritymagazine.com/articoli/lautenticazione-che-si-evolve-dalla-password-ai-token-u2f-parte-2/