IDEA CREATIVA: SINGLE-SIGN-ON CON G-SUITE PER I CLIENT DI SVILUPPO!

GSuite | Single-Sign-On (SSO)

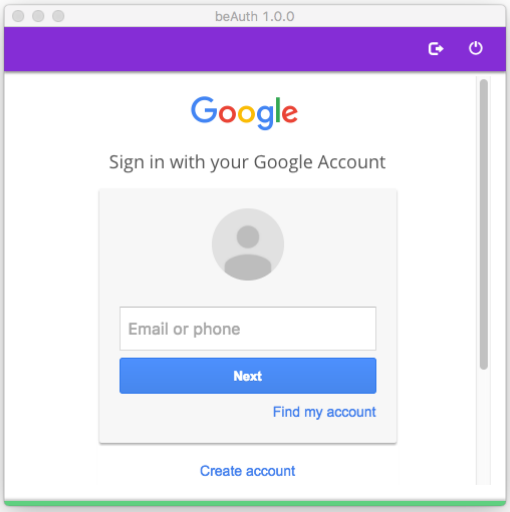
Simone Merlini | 4 Maggio 2017

Nell'ultimo articolo abbiamo parlato di come utilizzare gli account **G-suite** aziendali per effettuare il **Single-Sign-On** sulla web console di **Amazon Web Services**.

L'accesso alla web console copre solamente una parte delle esigenze di chi lavora quotidianamente con AWS. In particolare gli sviluppatori e i DevOps necessitano quasi sempre di una **coppia access key/secret key** sui loro personal computer per utilizzare la AWS CLI, per chiamare singole API di AWS (ad esempio quelle dei nuovi servizi di AI come Rekognition e Lex) e per poter utilizzare tutte quelle applicazioni desktop (pensiamo ad esempio ai vari file manager basati su S3 – come l'ottimo CloudBerry File Explorer, o ai client Git per l'utilizzo di CodeCommit) che a loro volta utilizzano le API di AWS.

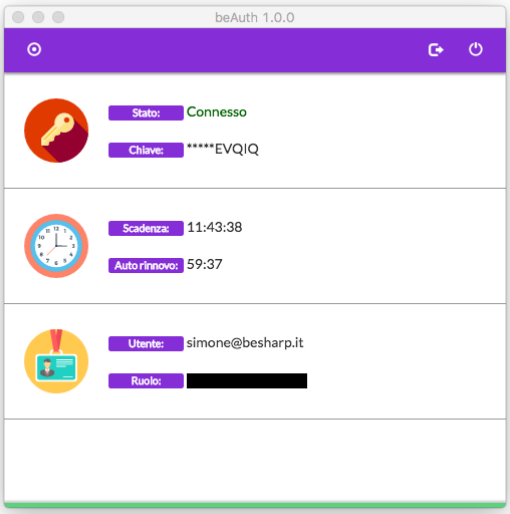
Access key e secret key **non sono associabili direttamente a uno IAM role** (il cui utilizzo tramite l'API AssumeRole abbiamo già visto essere una best practice di sicurezza), ma **necessitano di uno IAM user dedicato**, il che renderebbe di fatto vano l'assumere un ruolo AWS con delle credenziali centralizzate.

Questa limitazione necessitava giocoforza una soluzione un po' creativa ed è così che in beSharp **ci siamo inventati beAuth**.



All'avvio il programma mostra la schermata di login di G-suite (alla quale può essere associata la two-factor-authentication di Google)

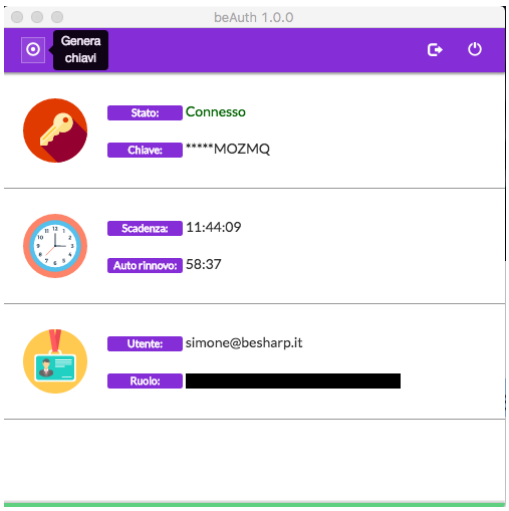
beAuth è un piccolo software installabile come agent all'interno del sistema operativo, che utilizza le credenziali G-suite e il meccanismo di SSO basato sul **protocollo SAML** (che abbiamo visto nel precedente articolo) per generare coppie access key/secret key temporanee e legate allo IAM role assunto, che vengono ruotate a intervalli prestabiliti (tipicamente 1 ora) all'interno della configurazione della AWS CLI. In questo modo, oltre alla CLI stessa, tutti i servizi che vi si appoggiano possono accedere alle risorse AWS ereditando temporaneamente i permessi del ruolo assunto, senza la necessità di disporre di uno IAM user dedicato.



Il pannello di controllo di beAuth mostra le informazioni relative alla sessione (che dura di default 12 ore) e alla durata della coppia access/secret key (che scade dopo un'ora), identifica l'utente connesso, lo IAM role assunto e le chiavi attive in un determinato momento. La chiave attiva viene configurata in automatico come default nella AWS CLI

Inoltre, siccome uno IAM role può essere assunto anche **cross-account**, con le stesse credenziali G-suite (opportunamente configurate) si possono ottenere coppie access/secret key per account differenti, **soluzione estremamente utile nel caso in cui l'azienda disponga di più account AWS** (ad esempio test – staging – produzione) oppure nel caso in cui si amministrino molteplici account AWS per conto di diversi clienti.

Per tutti i software che non si basano sulla CLI ma che necessitano l'inserimento diretto di coppie access/secret key, **beAuth permette di generare coppie "usa e getta"** della durata massima di 1 ora che possono essere inserite manualmente alla bisogna per singole chiamate o sessioni di lavoro.



Le chiavi possono essere rigenerate alla bisogna e copiate-incollate all'interno di qualsiasi applicazione di terze parti

I vantaggi di questa soluzione sono molteplici:

- **Praticità:** si può accedere a TUTTE le risorse AWS (e non solo alla CLI) con un unico account centralizzato, che coincide con l'account G-suite, senza la necessità di implementare complesse infrastrutture Active Directory.
- **Sicurezza:** il tutto si basa sul meccanismo di AssumeRole e STS token, quindi rispettando appieno le best practice di sicurezza dettate da AWS, le stesse che stanno alla base di servizi come S3, considerati sicuri da molte aziende della lista Fortune500. Il fatto che ci sia la possibilità di ruotare le chiavi con molta frequenza rende il tutto ancora più rock solid!
- **Semplicità:** beAuth si configura in automatico, per cui, una volta fatto il setup iniziale (5 minuti), tutto quello che è richiesto all'utente è di loggarsi con le proprie credenziali G-suite, senza dover gestire complicati script o switchare tutto il giorno tra credenziali e account differenti, potendosi così concentrare sulle proprie attività "core".
- **Portabilità:** beAuth è realizzata in Electron, quindi è portabile su qualunque sistema operativo desktop (Windows, Mac OS, Linux), ha un footprint ridottissimo in termini di consumo di risorse di sistema, è completamente sotto il controllo dell'utente e non necessita di permessi particolari per essere eseguita
- **Costi:** tutto questo genera 0 (ZERO) costi lato AWS, il che non guasta 😛

Che ne dite di questa nostra soluzione creativa? Siete interessati a implementare beAuth all'interno della vostra azienda? Volete realizzare qualcosa di simile in casa? Avete suggerimenti o domande?

Contattateci e commentate qui sotto per avere tutte le risposte!

Simone Merlini

CEO e co-fondatore di beSharp, Cloud Ninja ed early adopter di qualsiasi tipo di soluzione *aaS. Mi divido tra la tastiera del PC e quella a tasti bianchi e neri; sono specializzato nel deploy di cene pantagrueliche e nel test di bottiglie d'annata.

Get in touch

beSharp.it
proud2becloud@besharp.it