



**ELIMINA LE PERICOLOSE «CHIAVI MAESTRE»
CREANDO UN SISTEMA DI GESTIONE DELLE CHIAVI SSH**



www.splashtop.com/foxpass



COSA SONO LE CHIAVI MAESTRE?

In molte organizzazioni, gli ingegneri ottengono un accesso illimitato a ogni server e a tutti i dati che contiene. Si parla spesso di accesso di livello «amministratore» o «root». Un accesso così privo di vincoli è considerato una «chiave maestra», proprio come la «chiave maestra» di un edificio per uffici apre ogni porta.

E così come perdere la «chiave maestra» di un edificio per uffici può avere conseguenze disastrose, dare agli ingegneri «chiavi maestre» per server e dati può essere ancora più catastrofico se una di esse viene persa, rubata o usata in modo improprio.

PERCHÉ LE CHIAVI MAESTRE SONO UN ERRORE?

CASO 1: INSIDER MALINTENZIONATO – IMPROBABILE MA LETALE

Un insider malintenzionato è un dipendente che dispone sia dei mezzi necessari per accedere alle risorse di un'azienda, sia del movente per causare danni. Questo caso è altamente improbabile, poiché l'accesso root viene concesso solo ai dipendenti che hanno acquisito un certo livello di fiducia all'interno dell'organizzazione, ma non è uno scenario impossibile.

Si pensi, ad esempio, all'insider malintenzionato che ha sottratto gigabyte di dati a Tesla e li ha trasferiti a terzi. Secondo uno studio, oltre il 49% delle aziende è preoccupato per gli attacchi malevoli, poiché si tratta di una minaccia reale e in crescita.

CASO 2: INSIDER NEGLIGENTE – PROBABILE E NON TRACCIABILE

Nonostante tutta la formazione impartita ai professionisti IT, la maggior parte delle aziende resta vulnerabile ed è probabile che cada vittima di phishing e di altre forme di attacchi di social engineering. Secondo uno studio, infatti, oltre il 34% di tutte le violazioni del 2018 è stato causato da insider. Inoltre, questo tipo di attacchi di solito non è tracciabile e l'infrastruttura di un'azienda potrebbe essere già stata compromessa prima che qualcuno se ne accorga.

CASO 3: CREDENZIALI RUBATE – PROBABILE E TRACCIABILE

Anche il furto di laptop o di credenziali è un probabile attacco esterno, rivolto ai dipendenti con i diritti di accesso più elevati, ossia gli ingegneri e il team interno. Una volta ottenute queste credenziali, un attaccante esterno può accedere facilmente all'infrastruttura dell'azienda. Se le credenziali erano condivise tra più dipendenti, devono essere cambiate immediatamente, impedendo a tutti gli altri dipendenti di accedere ai server finché non potranno essere informati in modo sicuro delle nuove credenziali. Se ogni persona avesse le proprie credenziali per il server, basterebbe reimpostare solo quelle della persona interessata.

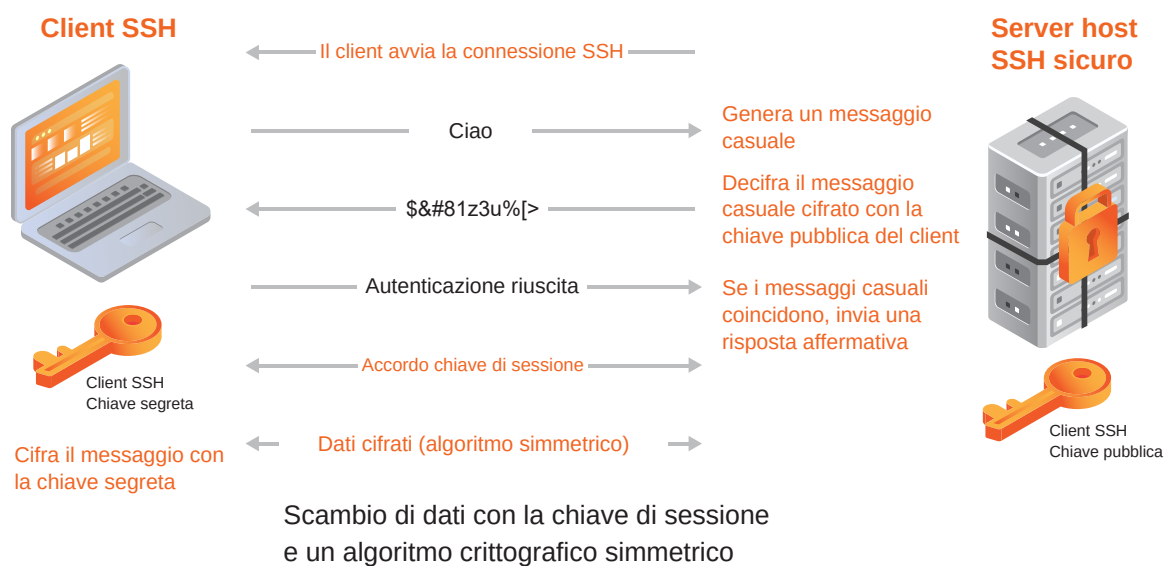


COME RIDURRE IL RISCHIO DI VIOLAZIONI DEI DATI?

Le aziende tecnologicamente all'avanguardia conoscono i rischi degli attacchi interni e limitano l'accesso dei propri ingegneri in base ai ruoli. Il controllo degli accessi basato sui ruoli (RBAC) limita l'accesso degli utenti in base al loro ruolo all'interno dell'organizzazione.

Adottare l'RBAC significa che un utente ha accesso solo a ciò di cui ha bisogno di frequente per svolgere il proprio lavoro. Tuttavia, «di frequente» non significa necessariamente ogni giorno, ogni settimana o anche ogni mese. L'opzione più sicura è concedere agli ingegneri il livello di accesso più basso possibile. Quando serve un accesso di livello superiore, questo dovrebbe essere monitorato, controllato e limitato nel tempo.

Come si ottiene un meccanismo di controllo degli accessi ben monitorato, controllato e limitato nel tempo? La soluzione migliore è un sistema in cui ogni ingegnere ha la propria chiave e che consente l'accesso a un server per una durata prestabilita: più breve è, meglio è.



QUANDO DOVREBBERO SCADERE I PERMESSI?

In genere, un ingegnere riceve i permessi dal momento in cui entra in un'azienda fino a quando la lascia. Un'opzione più dinamica, tuttavia, sarebbe concedere i permessi:

- Solo durante un turno di reperibilità
- Fino alla chiusura di un ticket di tipo Jira
- Solo se il permesso è stato utilizzato almeno una volta nell'ultimo mese

COS'È UNA CHIAVE SSH?

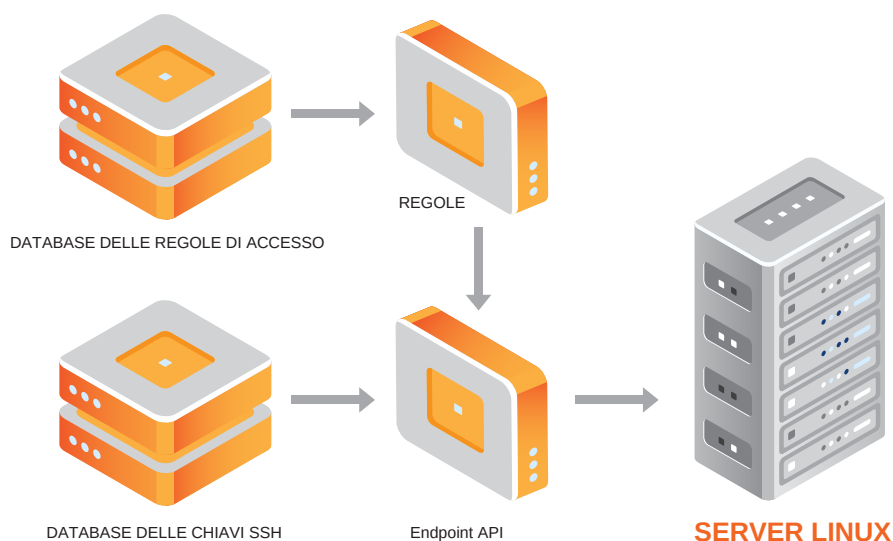
Una «chiave SSH» è un meccanismo crittograficamente sicuro per dimostrare la propria identità a un server quando si accede tramite la rete. Sotto questo aspetto è simile a una password lunga, ma le somiglianze finiscono qui. Una «chiave SSH» è in realtà composta da due parti: una chiave privata in un file e una chiave pubblica in un altro. Il contenuto dei due file è molto diverso, ma si completano a vicenda e vengono quindi creati contemporaneamente.

La metà privata della chiave SSH va custodita e non deve mai, ad esempio, lasciare il sistema su cui è stata generata. La metà pubblica, invece, può essere condivisa liberamente. La chiave privata non può essere ricostruita a partire dalla chiave pubblica. Copiando la parte pubblica della chiave su qualsiasi server remoto a cui l'utente deve poter accedere, il proprietario della metà privata otterrà l'accesso.

COME CREARE UN SISTEMA DI GESTIONE DELLE CHIAVI SSH?

I componenti necessari per creare un sistema di gestione delle chiavi SSH sono:

-  1 Un database in cui sono archiviate le parti pubbliche delle chiavi SSH degli utenti
-  2 Un database di regole di accesso che indica quali utenti accedono a quali server
-  3 Un server applicativo che supporta semplici applicazioni web ed endpoint API



COME CONFIGURARLO:

- 1 Tradizionalmente, il daemon sshd cerca su disco il file `authorized_keys` dell'utente nella sua directory `~/.ssh`.

In questa configurazione, invece, si sfrutta la direttiva sshd `AuthorizedKeysCommand` (disponibile in OpenSSH dalla versione 6.2) per indicare a sshd di usare i risultati dello script che si trova in `/usr/local/bin/get_keys.sh`. A tale scopo, modificare il percorso nel file `sshd_config`. Si noti che, in caso di errore, sshd ripiega comunque sul file `authorized_keys` presente su disco. Per farlo, modificare il file `sshd_config` come segue:

```
ubuntu@host:~$ vi /etc/ssh/sshd_config

[. . .]
AuthorizedKeysCommand /path/to/get_keys.sh
AuthorizedKeysCommandUser nobody
```

- 2 Lo script `/usr/local/bin/get_keys.sh` può essere scritto in qualsiasi linguaggio, ma si consiglia uno script Shell o Python. Il primo e unico argomento di questo script è il nome utente Linux dell'utente che richiede le chiavi. Lo script deve inoltrare la stessa query all'endpoint API realizzato. A tale scopo, utilizzare il comando `curl`, ad es. ``curl https://<endpoint>/get_keys/?user=$1``. Di seguito è riportato un esempio di `get_keys.sh` (specifico per AWS):

```
#!/bin/sh

user="$1"
hostname=`hostname`
instance_id=`curl -s -q -f
http://169.254.169.254/latest/meta-data/instance-id`
curl -s -q -f -m 5 "https://keyserver.example.com/sshkeys/?
user=${user}&hostname=${hostname}&instance_id=${instance_id}" 2>/dev/null

exit $?
```

Esempio di richiesta:

`https://keyserver.example.com/sshkeys/?user=aren&hostname=prod-db-1&instance_id=i-0b263919b6498b123`



Creare l'endpoint /sshkeys/ nell'API in modo che:

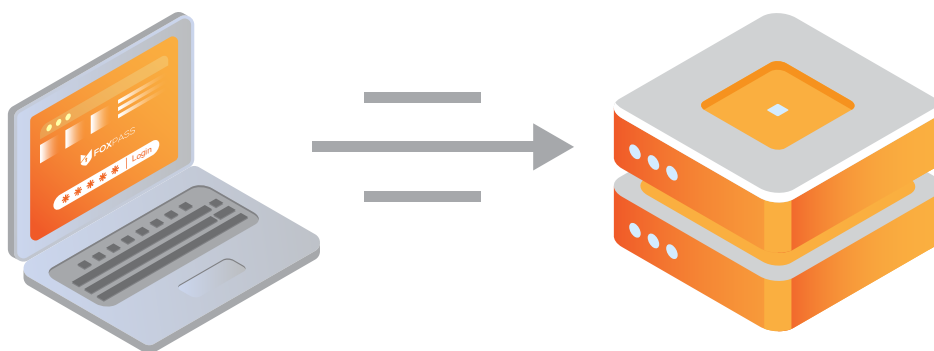
- ricavi l'argomento username dalla query string;
- cerchi le chiavi SSH di quell'utente nel database e
- restituisca le chiavi SSH nello stesso formato di un file authorized_keys: ogni chiave pubblica SSH su una riga a sé.

Esempio:

```
def sshkeys(request):  
    hostname = request.GET['hostname']  
    aws_instance_id = request.GET['aws_instance_id']  
    username = request.GET['user']  
  
    user = get_user(username)  
  
    keys = []  
    if is_user_allowed_on_host(user, hostname, aws_instance_id):  
        keys = user.keys  
  
    return HttpResponse("\n".join(key_text) for key in keys),  
    content_type='text/plain')
```



Creare un'applicazione web che gli utenti possano usare per gestire le chiavi SSH. È facile autenticare un utente nell'applicazione web tramite OAuth con GSuite o Office 365. L'utente deve poter aggiungere una nuova chiave e disattivarne una vecchia.



Semplice interfaccia web per l'accesso degli utenti (OAuth)
e il caricamento delle chiavi SSH (metà pubblica)

DATABASE DELLE CHIAVI SSH

Questa configurazione di base può essere ampliata introducendo un database di regole di accesso che disciplinano e controllano l'accesso ai server. Grazie all'integrazione con API esterne, l'accesso può essere modificato in base a determinate condizioni e ridotto o revocato di conseguenza.

- Per garantire che vengano distribuite solo le chiavi SSH dei dipendenti attuali: effettuare un controllo incrociato con le API di GSuite, Office365, Okta o OneLogin
- Per imporre la rotazione delle chiavi SSH: escludere quelle più vecchie di 90/180/360 giorni
- Per limitare i server a cui un utente può accedere: verificare i diritti di accesso definiti per ciascun utente. Se non esiste il permesso per quel server specifico, impedire all'utente di accedere al sistema

Esempio:

<https://keyserver.example.com/sshkeys/?user=aren&hostname=prod-db-1>

```
hostname=prod-db-1
```

```
«Macchina di produzione» match regex = ^prod-.*
```

```
L'utente «aren» ha ora diritti su una «Macchina di produzione»?
```

oppure

```
hostname=prod-db-1
```

```
«Macchina database» match regex = -db-\d+$
```

```
L'utente «aren» ha ora diritti su una «Macchina database»?
```

CONCLUSIONE

Le possibilità di utilizzo di questa configurazione per la gestione delle chiavi SSH e il controllo degli accessi sono infinite. Le vulnerabilità di sicurezza e gli attacchi interni sono sempre una grande preoccupazione per le organizzazioni che possiedono dati sensibili condivisibili con gli ingegneri, poiché una loro fuga può avere un impatto enorme. Ha sempre senso concedere ai dipendenti i privilegi minimi e limitare l'accesso alle risorse critiche quando viene concesso. Se Ashley Madison, Target o Tesla avessero avuto un controllo degli accessi migliore, forse non avrebbero subito violazioni della sicurezza così costose e pubbliche. La gestione delle chiavi SSH è un meccanismo utile per proteggere l'infrastruttura e aiutare a impedire che i team interni divulgino all'esterno informazioni riservate.

COME FOXPASS PUÒ AIUTARTI

Foxpass offre prodotti completi e facili da usare per la gestione delle chiavi SSH e il controllo degli accessi privilegiati, che consentono agli ingegneri di accedere ai server necessari per un periodo di tempo predefinito. Inoltre, Foxpass è un sistema completo di gestione di utenti e gruppi per Linux che si sincronizza con le directory più diffuse come GSuite, Office365 e Okta.