# Opkssh - ssh via OIDC

Visto che abbiamo installato Authentik, utilizziamolo per creare i gruppi di accesso ai server via ssh. Per farlo utilizzeremo il progetto alla pagina (https://github.com/openpubkey/opkssh)[https://github.com/openpubkey/opkssh].

## Passo 1: Configurazione del provider e applicazione

Anzitutto entriamo nell'interfaccia di amministrazione di Authentik, e creiamo un provider ad hoc (Menù di sinistra -> Applicazioni -> Providers): dovrà essere di tipo *public* e **NON** confidential. Segnamoci il client-id

- Provider Name: opkssh-provider
- Tipologia client: Pubblico
- Reindirizzamento URI
    - http://localhost:3000/login-callback
    - http://localhost:10001/login-callback
    - http://localhost:11110/login-callback Esatto sono 3

L'ultima voce da cambiare è la modalità del soggetto che è meglio sia *Basato sull'UUID dell'utente*. Il resto può essere lasciato così com'è.

Bisogna associarci un'applicazione (Menù di sinistra -> Applicazioni -> Applicazioni):

- Application name: Opkssh
- Slug: opkssh
- Provider: opkssh-provider

Il resto, as is.

## Passo 2: Creazione del gruppo (almeno uno)

Per entrare sui server, a meno che non si vogliano dare i permessi nominativi su ogni macchina, possiamo abilitare gruppi di persone a loggarsi come l'utente "X" (**che dovrà essere presente sulle macchine come utente locale**). Sei l'utente "Mario Rossi"? Se fai parte del

gruppo "ssh-admins" allora potrai loggarti sulla macchina come suo utente locale "X" (X può essere un qualsiasi utente di sistema, persino root). Va da sé che possiamo creare quanti gruppi vogliamo per questo scopo: magari il gruppo *devops*, il gruppo *secops*, il gruppo *sparecchiavo* a seconda di quanto vogliamo essere granulari sul parco host a disposizione. Noi creeremo solo il *ssh-admins*.

Detto questo: Menù di sinistra -> Cartelle -> Gruppi e creiamo *ssh-admins*. Dopo questo, menù in alto, Utentt -> Aggiunti utente esistente e diamogli l'utente che vogliamo testare.

**ATTENZIONE**: per onore di cronaca sarebbe il caso di associare l'applicazione *Opkssh* testé creata ai soli aventi diritto alle connessioni ssh, per cui forse è il caso di creare un gruppo *ssh-access* e dei sottogruppi *ssh-admins*, *ssh-devops*, etc in modo da associarle *ssh-access*. In questo momento soprassiedo, perché ho sonno.

## Passo 3: Installazione lato server

Procediamo con l'installazione lato server:

```
wget -qO- "https://raw.githubusercontent.com/openpubkey/opkssh/main/scripts/install-linux.sh" | sudo bash
```

Modifichiamo il file `/etc/ssh/sshd_config`

Sostituiamo

```
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```

con

```
AuthorizedKeysCommand /usr/local/bin/opkssh verify %u %k %t
AuthorizedKeysCommandUser opksshuser
```

e riavviamo il server con `systemctl restart ssh`.

Modifichiamo il file `/etc/opk/providers` affinché contenga solo il nostro identity provider:

```
https://auth.example.org/application/o/opkssh/
AZEQOFmwX6qTQOG63ekDHMceSFehjkRUdtayA3yh 24h
```

dove `AZEQOFmwX6qTQOG63ekDHMceSFehjkRUdtayA3yh` è il client id creato al passo 1, in fase di configurazione del provider.

Creiamo l'utente che verrà utilizzato da chi si logga via oidc:

```
adduser oidcuser
info: Adding user `oidcuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `oidcuser' (1003) ...
info: Adding new user `oidcuser' (1003) with group `oidcuser (1003)' ...
info: Creating home directory `/home/oidcuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for oidcuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `oidcuser' to supplemental / extra groups `users' ...:
info: Adding user `oidcuser' to group `users' ...

# usermod -aG sudo oidcuser
```

e specifichiamo che chi fa parte del gruppo `ssh-admins` dell'identity provider, allora può loggarsi nella macchina (e entrerà come utente `oidcuser`)

```
# opkssh add oidcuser oidc:groups:ssh-admins https://auth.example.org/
application/o/opkssh/
2026/03/20 09:22:23 Successfully added user with email oidc:groups:ssh-admins
with principal oidcuser to the policy file
Successfully added new policy to /etc/opk/auth_id
```

## Passo 4: Tlog

Per tenere traccia di che cosa faccia l'utente *oidcuser* utilizzeremo *tlog*.

```
sudo apt update && sudo apt install tlog jq -y
```

Configuriamo la registrazione:

```
sudo mkdir -p /etc/tlog
sudo tee /etc/tlog/tlog-rec-session.conf <<EOF
{
  "shell": "/bin/bash",
  "writer": "file",
  "file": {
    "path": "/var/log/tlog-ssh.log"
  }
}
EOF
```

Configuriamo il player:

```
sudo tee /etc/tlog/tlog-play.conf <<EOF
{
  "reader": "file",
  "file": {
    "path": "/var/log/tlog-ssh.log"
  }
}
EOF
```

Prepariamo i file di log e permessi:

```
# Crea il file di log e imposta i permessi (Scrivibile da tutti, ma non cancellabile)
sudo touch /var/log/tlog-ssh.log
sudo chmod 622 /var/log/tlog-ssh.log

# Configura la directory temporanea per i lock (evita l'errore Permission Denied)
sudo tee /etc/tmpfiles.d/tlog.conf <<EOF
d /run/tlog 1777 root root -
```

```
EOF
sudo systemd-tmpfiles --create /etc/tmpfiles.d/tlog.conf
```

Abilitiamo la registrazione per l'utente:

```
# Aggiungi tlog alle shell valide
echo "/usr/bin/tlog-rec-session" | sudo tee -a /etc/shells

# Cambia la shell dell'utente (es. oidcuser)
sudo chsh -s /usr/bin/tlog-rec-session oidcuser
```

### Per l'amministratore

Ecco una cheasheet per un amministratore:

- Vedere quante connessioni ci sono: `sudo cat /var/log/tlog-ssh.log | jq -r '.rec' | sort | uniq | wc -l`
- Elencare le sessioni (Utente, ID, Data): `sudo cat /var/log/tlog-ssh.log | jq -r '[.user, .rec, (.time | strftime("%Y-%m-%d %H:%M:%S"))] | @tsv' | column -tH`
- Riprodurre l'ultima sessione: `tlog-play`
- Riprofurre una sessione specifica: `tlog-play -M "rec=INSERISCI_ID_QUI"`

## Passo 5: Installazione del client

Il client è quello che richiede il token all'IdP: una volta ottenuto, riceverà la chiave ecdsa che verrà scritta sotto *~/.ssh/* e potrà essere utilizzata per connettersi ai server remoti. Il file di installazione è prelevabile all'indirizzo (https://github.com/openpubkey/opkssh)[https://github.com/openpubkey/opkssh], nella sezione releases.

Una volta installato sarà sufficiente, da utente:

```
opkssh login --create-config
```

Questo creerà `~/.opk/config.yml`. Cancelliamone il contenuto e rimpiaziamolo:

```
---
default_provider: webchooser
```

```
providers:
  - alias: Dreadful
    issuer: https://auth.example.org/application/o/opkssh/
    client_id: AZEQOFmwX6qTQOG63ekDHMceSFehjkRUdtayA3yh
    scopes: openid profile email offlince_access
    access_type: offline
    prompt: consent
    redirect_uris:
      - http://localhost:3000/login-callback
      - http://localhost:10001/login-callback
      - http://localhost:11110/login-callback
```

Ancora una volta, `AZEQOFmwX6qTQOG63ekDHMceSFehjkRUdtayA3yh` è il client id creato al passo 1, in fase di confiurazione del provider.

Giunti a questo punto, per l'autenticazione sarà sufficiente eseguire un

```
opkssh login
```

si aprirà il browser e chiederà le credenziali. Da questo momento, potremo iniziare a connetterci tramite

```
ssh oidcuser@server
```

---

[Che il talpone sia con voi]

[EOF]

Fluxer Icon

Canale di discussione e supporto

Canale di discussione e supporto. Per confrontarti su configurazioni o altro entra su fluxer.dreadful.work (https://fluxer.dreadful.work/invite/dreadful).