Using SSH with a Password

Preface

Normally you will use SSH public key authentication to connect to Dirvish clients. However there are cases where this is not possible, for example on webhosting packages. You might be lucky to have SSH access with password authentification at all, hopefully along with rsync so a Dirvish backup is possible.

This instructions were tested with a Dirvish server on debian 10 (buster), and a webgo.de webhosting package „pro“. But basically it should work with any linux distro on server side and any webhosting package that includes a SSH access and rsync, probably with small modifications.

The sample parameters used are:

- Client Host: s999.goserver.host
- Client User: web999
- Client SSH Password: ssh_passwd
- Dirvish bank: /backup/

Test SSH and set up sshpass

First do a normal SSH from your Dirvish server to the client. Accept adding the client to the list of known hosts:

```
# ssh web999@s999.goserver.host
...
Are you sure you want to continue connecting (yes/no)? yes
...
web999@s999.goserver.host's password: ssh_passwd
...
web999 # exit
#
```

Install sshpass:

```
# apt install sshpass
```

Save ssh password into a file:

```
# echo 'ssh_passwd' > /etc/dirvish/web999.pass
# chmod 0400 /etc/dirvish/web999.pass
```

Test if working so far:

```
# sshpass -f /etc/dirvish/web999.pass ssh web999@s999.goserver.host
```

```
web999 # exit
#
```

Make Dirvish use sshpass

Edit your clients Dirvish config file, and add the `rsh` parameter as:

```
# mkdir -p /backup/web999/dirvish
# nano /backup/web999/dirvish/default.conf
---
client: web999@s999.goserver.host

tree: ~/www/

exclude:
    trash/**
    cache/**
    tmp/**

rsh: sshpass -f /etc/dirvish/web999.pass ssh -l web999
```

Add web999 as vault under _Runall_ in /etc/dirvish/master.conf.

You should now be able to run your init command:

```
# dirvish --vault web999 --init
```

Backing up the databases too

Typically webhosting packages also include mysql (mariadb) databases. If you are lucky to have access to the mysql and mysqldump programs in SSH, you can save your databases too.

The sample parameters used are:

- MySQL User: web999
- MySQL Password: sql_passwd

Create pre-client script:

```
# nano /backup/web999/dirvish/pre-client
---
#!/bin/bash

for db in $( mysql -uweb999 -psql_passwd -e 'show databases' -s \
    | grep -v 'Database' \
    | grep -v 'mysql' \
    | grep -v 'information_schema' \
```

```
    | grep -v 'performance_schema' \
)
do
    mysqldump -uweb318 -psql_passwd --default-character-set=utf8mb4 --no-
tablespaces -B $db > ~/www/$db.sql
done
```

Create post-client script:

```
# nano /backup/web999/dirvish/post-client
---
#!/bin/bash

for db in $( mysql -uweb999 -psql_passwd -e 'show databases' -s \
    | grep -v 'Database' \
    | grep -v 'mysql' \
    | grep -v 'information_schema' \
    | grep -v 'performance_schema' \
)
do
    rm -f ~/www/$db.sql
done

rm -f ~/www/dirvish.pre-client
rm -f ~/www/dirvish.post-client
```

Make scripts executable:

```
# chmod 755 /backup/web999/dirvish/pre-client
# chmod 755 /backup/web999/dirvish/post-client
```

Finally update the config file to use the scripts as:

```
# nano /backup/web999/dirvish/default.conf
---
client: web999@s999.goserver.host

tree: ~/www/

exclude:
    dirvish.pre-client
    dirvish.post-client
    trash/**
    cache/**
    tmp/**

rsh: sshpass -f /etc/dirvish/web999.pass ssh -l web999

pre-server: ; for prefix in pre post ; do rsync --rsh "sshpass -f
/etc/dirvish/web999.pass ssh -l web999" ../../dirvish/$prefix-client
$DIRVISH_CLIENT:~/www/dirvish.$prefix-client ; done ; exit 0
```

```
post-server: ; exit 0

pre-client: ; ~/www/dirvish.pre-client

post-client: ; ~/www/dirvish.post-client
```

From:
https://wiki.diala.net/ - **Diala Wiki**

Permanent link:
https://wiki.diala.net/doc:sshpass

Last update: **27.08.2023 18:02**