Lab~Secure Your Linux Desktop and SSH Login Using Two Factor Google Authenticator

We will enable two factor authentication in out ubuntu server. To implement that we are going to use multifactor authentication with Google Authenticator.

Step 1: Install Google Authenticator from following link in your Android device/iPhone/iPad/BlackBerry/Firefox devices:

```
https://support.google.com/accounts/answer/1066447?hl=en
```

Step 2: Install Google Authenticator in your Ubuntu

```
# sudo apt-get install libpam-google-authenticator
```

Step 3: Create an Authentication Key

Log in as the user you'll be logging in with remotely and run the google-authenticator command to create a secret key for that user.

```
$ google-authenticator

Do you want authentication tokens to be time-based (y/n) y
```

You will get some QR code ouput like bellow:

You will be prompted for some configurations.

1. Scan the QRcode that appears with the Google Authenticator app or you can add the secret key

Google Authenticator app.

2. Save the backup codes listed somewhere safe. They will allow you to regain access if you lose your phone with the Authenticator app.

3. Next it will ask several question; unless you have a good reason to, the defaults presented are sane. Just enter "y" for them.

```
Do you want me to update your "/home/fakrul/.google_authenticator" file (y/n)

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n)

By default, tokens are good for 30 seconds and in order to compensate for
possible time-skew between the client and the server, we allow an extra
token before and after the current time. If you experience problems with poor
time synchronization, you can increase the window from its default
size of 1:30min to about 4min. Do you want to do so (y/n)

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting (y/n)
```

Step 4: Activate Google Authenticator

Enable Google Authenticator for SSH logins.

```
# sudo vi /etc/pam.d/sshd
auth required pam_google_authenticator.so
```

Next, open the /etc/ssh/sshd_config file, locate the ChallengeResponseAuthentication line, and change it to read as follows.

```
# vi /etc/ssh/sshd_config
ChallengeResponseAuthentication yes

# sudo service ssh restart
```