How to Secure Your Instances with Multi-factor Authentication centos

### **Install Google Authenticator**

Update the Ubuntu repositories to download the latest version of the authenticator:

* sudo apt-get update

Now that your repositories are up to date, install the latest version of the PAM module:

* sudo apt-get install libpam-google-authenticator

### **Enable Google Authenticator**

# google-authenticator

The application will ask you whether the authentication tokens should be time-based or not. Since we are going to set up time-based tokens, enter y to continue.

Do you want authentication tokens to be time-based (y/n) y  
The application will generate a QR code on your screen which you need to scan with the smartphone application. Run the Google Authenticator application on your smartphone and scan the QR code.

The application will generate a new secret key, verification code and emergency scratch codes too. Keep these codes in a safe place.

Next, the application will ask you whether to save the key and the settings in the .google\_authenticator file for your system user. Enter y to continue.  
  
Do you want me to update your “/root/.google\_authenticator” file? (y/n) y  
The next question is whether you want to disallow multiple uses of the same authentication token. Each token will be valid for 30 seconds and will expire immediately after the use. Enter y to continue.

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) y  
If you don’t expect to have syncing issues between the client and the server, answer the next question with n.

By default, tokens are good for 30 seconds. 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 +-1min (window size of 3) to about +-4min (window size of 17 acceptable tokens).  
Do you want to do so? (y/n) n  
Next, you can enable rate-limiting for the authentication module. This will allow you to have 3 login attempts every 30 seconds so enter y to enable this feature.  
  
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) y  
By answering this question you are done with the Google Authenticator setup. The settings are saved in the ~/.google\_authenticator file and if you want to create a backup of this file you can copy it to a safe location.

### **Configure openssh (SSHD)**

# vi /etc/pam.d/common-auth

Add the highlighted line at the end of the file:

/etc/pam.d/common-auth

...

# and here are more per-package modules (the "Additional" block)

session required pam\_unix.so

session optional pam\_systemd.so

# end of pam-auth-update config

auth required pam\_google\_authenticator.so nullok

Save and close the file after adding the line.

In the case of a headless server, such as a DigitalOcean Droplet, you will edit the /etc/pam.d/common-session file instead. Open the relevant file based on your environment:

* sudo nano /etc/pam.d/common-session

Add the highlighted line to the end of the file:

/etc/pam.d/common-session

#

# /etc/pam.d/common-session - session-related modules common to all services

#

...

# # and here are more per-package modules (the "Additional" block)

session required pam\_unix.so

session optional pam\_systemd.so

# end of pam-auth-update config

auth required pam\_google\_authenticator.so nullok

change the sshd config file

# vi /etc/ssh/sshd\_config

Permitrootlogin no >> yes

Passwordauthentication no >>yes

ChallengeResponseAuthentication yes

Save and exit

And restart the sshd service

#service sshd restart

Once done try to login to server .

#ssh root@locathost

<https://www.digitalocean.com/community/tutorials/how-to-configure-multi-factor-authentication-on-ubuntu-18-04>

**For creating user to setup MFA follow the below steps**

#adduser test

#passwd test

# su – test

$google authenticator

The application will ask you whether the authentication tokens should be time-based or not. Since we are going to set up time-based tokens, enter y to continue.

Do you want authentication tokens to be time-based (y/n) y  
The application will generate a QR code on your screen which you need to scan with the smartphone application. Run the Google Authenticator application on your smartphone and scan the QR code.

The application will generate a new secret key, verification code and emergency scratch codes too. Keep these codes in a safe place.

Next, the application will ask you whether to save the key and the settings in the .google\_authenticator file for your system user. Enter y to continue.  
  
Do you want me to update your “/root/.google\_authenticator” file? (y/n) y  
The next question is whether you want to disallow multiple uses of the same authentication token. Each token will be valid for 30 seconds and will expire immediately after the use. Enter y to continue.

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) y  
If you don’t expect to have syncing issues between the client and the server, answer the next question with n.

By default, tokens are good for 30 seconds. 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 +-1min (window size of 3) to about +-4min (window size of 17 acceptable tokens).  
Do you want to do so? (y/n) n  
Next, you can enable rate-limiting for the authentication module. This will allow you to have 3 login attempts every 30 seconds so enter y to enable this feature.  
  
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) y  
By answering this question you are done with the Google Authenticator setup. The settings are saved in the ~/.google\_authenticator file and if you want to create a backup of this file you can copy it to a safe location.

Install the app and scan the bar code

Next to login to server

Open new window byputting the ip and click enter

Give the user name : test

Here it will ask the password authentication 2 times put that you will be able to login to server.