Two Factor Authentication

# Overview

A common method to secure password authentication is by using Two Factor Authentication (2FA) where a user has to both enter their password, and confirm their identity using another process such as text, email, or google authenticator.

# Mobile - Ruby on Rails

## Overview

Moblie 2FA sends a client a sms message with a code after password login, the client then submits the password to the server in a separate form and the server checks for validity with the sms provider to confirm a users identity. There are many sms providers available for a developer to use, a popular option is twilio.

## Twilio

Twilio provides a range of services which allow a developer to send messages in various different formats. For 2FA they provide a twilio verify api service which sends a random verify code, then the server can verifiy with the api that the client has received and submitted the correct password.

Prerequistes:

* Create a twilio account, then on the dashboard create a new verify service with application name.
* Add the 'twilio-ruby' gem to the gemfile.
* Make sure there is a mobile number entry in the database for User.
* Add twilio credentials to rails credentials (account details and service sid)
* Add mobile number to user seed

Start by making a new verification method in the sessions controller and associtated rails route:

* Upon sucessful password login assign user id to session[:two\_factor\_auth\_id] and redirect to new 2FA page, on redirect automatically send verification request to user mobile number using twilio gem
* Redirect user if they do not have a session[:two\_factor\_auth\_id]
* 2FA page will then have input box (http post request form) for the client to enter the auth code which they receive in a sms message

def two\_factor\_auth

if session[:two\_factor\_auth\_id]

client\_verify\_number = User.find\_by(id: session[:two\_factor\_auth\_id]).mobile\_number

client = Twilio::REST::Client.new(Rails.application.credentials.twilio[:account\_sid], Rails.application.credentials.twilio[:auth\_token])

client.verify

.services(Rails.application.credentials.twilio[:verify\_service\_sid])

.verifications

.create(to: client\_verify\_number, channel: 'sms')

render :two\_factor\_auth

else

redirect\_to(:login)

end

end

Then make the post request verfication check method in the sessions controller and associated route:

* Find user with session[:two\_factor\_auth\_id] and send verification check request to twilio
* If the code is correct twilio will return the 'approved' status and you can reset the session and then add the user\_id to the session cookie and redirect to, logging them in

def two\_factor\_auth\_verify

if session[:two\_factor\_auth\_id]

user = User.find\_by(id: session[:two\_factor\_auth\_id])

client = Twilio::REST::Client.new(Rails.application.credentials.twilio[:account\_sid], Rails.application.credentials.twilio[:auth\_token])

verification\_check = client.verify

.services(Rails.application.credentials.twilio[:verify\_service\_sid])

.verification\_checks

.create(to: user.mobile\_number, code: params[:auth\_code])

reset\_session

if verification\_check.status == 'approved'

session[:user\_id] = user.id

redirect\_to :admin, notice: "#{user.username} welcome back to your home-server!"

else

redirect\_to :login, notice: '2fa code incorrect, please try again'

end

else

redirect\_to(:login)

end

end