

FEPS

**Two-Factor Authentication
for External SSH Access**



**UNIVERSITY OF
SURREY**

Preface

To improve security for remote services in FEPS IT, two-factor authentication has now been made a requirement for access to external services. This document covers two different methods for generating your two-factor authentication methods to access the FEPS SSH gateway.

All access through SSH gateways will **require** a two-factor authentication method.

Background

Security is based around using any of three methods for identification:

Something you know – Your username, password, security questions

Something you have – A Soft/Hard token, public/private keypair, SMS token.

Something you are – Biometrics (retina scan, finger print, etc)

Normal security, such as logging into your computer or email account, requires two forms of ‘something you know’ - Your username and password. Unfortunately the ‘something you know’ is easy to steal as it can be intercepted during communication, or the user can be duped into entering the information into non-genuine sites. As the information rarely changes, the same information is often used across multiple accounts.

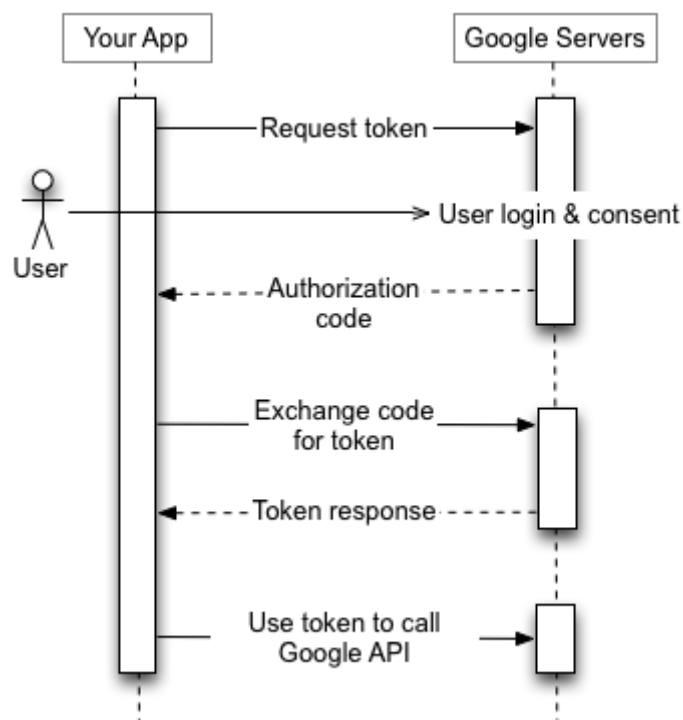
Some online systems use ‘Security Questions’ in order to improve this method of security. During account setup, the system will prompt you to enter answers to multiple security questions. During login, you will be prompted to enter the answer from a random question. However, it is still possible to steal this information. Questions like ‘What is your Mother’s maiden name’ are used by multiple services, and the answer never changes.

Most online accounts now offer forms of two-factor authentication. This will require ‘Something you know’. A good example of this will be your bank, which may require you to enter a code sent to you via SMS (SMS token) or verify with a code from a token device.

Rarely will you find a service using three-factor authentication, but biometrics are often used as a single factor authentication method as they are considered more secure (harder to steal, but not impossible) than Username and Password (Something you know).

Two-Factor Token

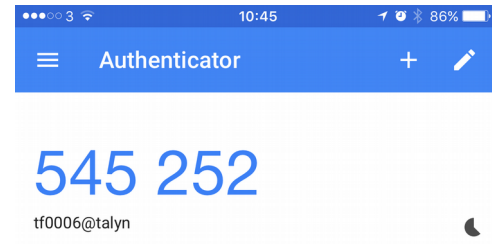
This uses your smartphone to hold a soft token, which is used to verify your identity. Once you have identified to the server with your username and password, the server will request you to enter a verification code. Your smartphone will display this code (which rotates every 30-seconds), which you can then enter into the prompt. The code is sent to the server which is then verified against Google's servers via an API call. If the code you entered is correct, then the server will allow you to log in. If the code does not match, then the server will reject your connection.



Two-Factor Token


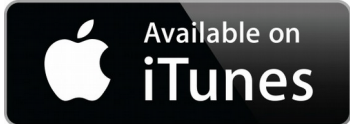





Prerequisites

- Android or Apple smart phone
- Authenticator App



You will need to be on campus at the University of Surrey and logged onto a FEPS Linux machine running Ubuntu 16.04 (Xenial) or CentOS 7.3.

Authenticator Apps

		
Google Authenticator 	Download Google Authenticator for Android (free)	Download Google Authenticator for iPhone/iPad (free)
LastPass Authenticator 	Download LastPass Authenticator for Android (free) *	Download LastPass Authenticator for iPhone/iPad (free) *
1Password 	Download 1Password for Android (subscription) **	Download 1Password for iPhone/iPad (subscription) **
Microsoft Authenticator 	Download Microsoft Authenticator for Android (free)	Download Microsoft Authenticator for iPhone/iPad (free)
Authenticator by Matt Rubin 	Unavailable	Download Authenticator for iPhone/iPad (free)

* = LastPass offers a free and subscription tier of their LastPass service – more information at <https://www.lastpass.com>

** = 1Password is only available as a paid subscription service. More information is available at <https://agilebits.com>

From your FEPS IT Linux machine:

1. Open a Terminal
2. From the command line or terminal, run the following command and press Enter:

```
google-authenticator
```

You will now be guided through the security settings for your Google Authenticator Two-Factor authentication. For all responses, enter y or n to signify Yes or No, then press 'Enter'.

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

Choose: 'Yes'

This will begin the token generation process

3. Your token will now be generated and displayed in the terminal. It will show the following:

```
Your QR Code – you will need to scan this using your Authenticator App. This will now
Your Secret Key – T
Your Verification Code -
Your emergency scratch codes -
```

Using your Smartphone and authenticator app, add a new account / entry, choosing the QR Code entry option. Now scan the QR Code displayed in the terminal. Your authenticator token will be added to your authenticator app. A 6-digit code will be displayed on the screen along with a 30-second countdown timer. The code will be reset every time the counter reaches 0-seconds.

Now continue with the steps below.

4. If you are replacing your existing token, follow this step, otherwise continue to step 5.

- Do you want me to update your "/user/<path_to_home>/<username>/google_authenticator" file (y/n)

If you are replacing / updating your token, you will first be asked if you want to update your existing .google_authenticator file. If you have not previously set up Google Authenticator, this will not be displayed.

NOTE: This will replace / invalidate / void your previous Google Two-Factor Authentication token!

Choose: 'Yes' if you want to replace your token.

5. You will now be asked a series of questions. Please enter the responses as documented under each question.

- 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)

Choose: 'Yes'

- 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)

Choose: 'No'

- 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)

Choose: 'Yes'

6. You have now successfully generated your two-factor authentication token. This token is not required for use when logging into your local machine.

It will be required when connecting via SSH to the External SSH gateway – external-ssh.eps.surrey.ac.uk .

Testing your Two-Factor token

You can test if your token is working by establishing an SSH connection to ssh-test01.eps.surrey.ac.uk

During login, you should be prompted for:

- Username
- Password
- Verification Code

NOTE: The Google Two-Factor authentication will be bypassed if a matching SSH keypair is present! In this instance you will just be prompted to enter the password for your keypair, and the two-factor challenge/response prompt will not be presented.