Twofactor Authentication

Enables a second authentication step after the default GUID auth. This is handy for games where GUID spoofing is possible (e.g. Call of Duty)

Prerequisites

`twofactorauth.enabled` set to `true` in `config/basics.json`.
An authenticator which implements IETF RFC 6238. See Appendix 1 for possible suggestions.

Usage

Set up your authenticator with your personal secret. You can find your secret in `config/users.json`.
To show the secret of others you can use `!secret NAME <name or pid>` (for online players) or `!secret GUID <guid>` <guid> is the 8 character version of the GUID

After joining the server you will have to authenticate yourself using the temporary password given by the authenticator.
For Example: `!login 162534`

Examples

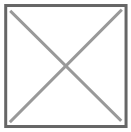
Setting up your authenticator

Example secret used: `2CB4SVARUIHVJ64J` (You have to replace this with your own secret if you follow this guide)

GAuth [1]:

Spoiler anzeigen

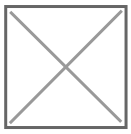
click the edit icon and then the **Add** button



Give it a sensible **Account name** and input your secret in the Secret key field and click the **Add** button



If you want u can delete the example alice@google.com account by clicking the **X** next to it.
It will look something like this:



The 6 digit number is your temporary password which you need for the `!login 490186` command on the server.

In the top right corner you will see the timer how long the password will be valid (there is about a 30 seconds padding before and after the timer to count for not perfectly synced clocks and slow input).

Appendix 1: List of authenticators

[1] GAuth: Web based authenticator http://gauth.apps.gbraad.nl/ (also available as Phone APP and Webapp)
[2] Google Authenticator: Phone App for Android, iOS and Blackberry (open with your phone)
[3] FreeOTP: Open Source alternative to Google Authenticator. Available for Android and iOS
[4] WinAuth: Open Source authenticator for Windows: https://winauth.com/

for more see: https://en.wikipedia.org/wiki/Time-base…implementations