Local Password Generator

I just read an article about rules for passwords and it made some sense but it glaring errors. I said you should generate a long passwords, at least 20 characters long. Ok that makes sense. Maybe a little short for the future. But it said that random passwords are too difficult to remember so you should use an XKCD type password of words strung together with hyphens.

- The article implied that you would have a 'my password' that you use on all accounts, since you need to remember it. Horrible.

Some other ideas from google search:

- In a list from UC Santa Barbara said 'create a password that's hard to guess but easy to remember'. Horrible.
- Never write your password down. Horrible.

Here's my personal guidelines for a password:

- use different passwords for different accounts (of course)
- passwords should be at least 24 characters long, preferable more. 30 is better.
- use a password that is hard/infeasible to crack (crytographically random), with letters, numbers and optionally symbols. No words or garbled words.
- use a password manager. don't try to remember your passwords unless you are one of those rare people who can recall Pi to hundreds of numbers.
- keep a printed list of my passwords in a secure location.

And of course you want to use MFA and passkeys if possible.

One thing missing is how to generate passwords that meet my criteria. I usually passwordgenerator.net. There are dozens of others on line. **But how do you know that the web app generating your password is not storing it somewhere? Saving it for some other use? You don't really know** unless you generate one locally (assuming your computer is secure lol).

Build and Test

For ease of use and paranoia, I decided to write my own password generator. To make it easy, I'm going to use AI code generation to help out. I am using Go since it has the right libraries and it can generate binaries for linux, windows and macos from the same source.

Creating the program

I am going to attempt to generate it using Aider and Anthropic Claude-3.5.sonnet. I use Aider on a terminal and VSCode with Supermaven AI for code completions.

Here's the prompt I will be using:

```
using the Go programming language, create a command line application that
generates one or more passwords with the criteria:
```

```
- has a command line argument -l[n] or --length=n to specify the length of
the password.  default to 24 if not specified.
- has a command line argument -c[n] or --count=n to specify the number of
passwords to generate. default to 1 if not specified.
- have a command line argument -s or --symbols to ask it to include symbols
in the password  By default the password should only include ASCII letters
and numbers.
- have a command line argument -h or --help to display a help message
describing the command line arguments.
- minimum length of 24 characters
- cryptographically random
- output the results to stdout with 2 lines per password: one line for the
complete password and a second line where the password is broken up into 3
character sections divided by a space. add a line feed between each pairs
of lines.
- use the Golang crypto/rand package to generate the passwords
```

The program builds and runs on linux with out modification. I added additional prompts:

- create a Makefile that builds main.go with options for linux, macOS or windows. the makefile shoul d default to building for linux
    - the Makefile had a warning so I fixed it by hand
- create a test file using the Golang test framework. test all options
    - the test file seems to check everything

Code Review

When I build with aider and anthropic, I like to do a code review using a different tool and model. In this case, I disable Supermaven and enable Codeium. I opened the Codeium window and gave it the prompt:

- review main.go for bugs, style and idiomatic golang.

Codeium did not find any problems. It gave some recommendations but they appeared to me to just be variations on the original code, no real changes. Just like I had two human programs write the same thing. They wouldn't be identical. So I did not accept any changes.

I reviewed the code by hand and did not find any problems. I focused on the generatePassword function to be sure it was generating random data. And it was.

Finally I switched back to aider and added a modification to the output:

- change the output to have 4 lines per password. the first line is blank, the second line is a stri ng of hyphens of lenght 80, and then existing output

Conclusion

Now I can generate my own passwords, print the output if I want to. But, I would never store the passwords in the clear on my computer(s). I would use a password manager to store them or just keep a printout handly.

Using this tool, I typically create a page full of the pregenerated passwords and keep it in a secure location. Then when I need to add a new password, I use the next one on the list, write something so I'll know what site/app it is used for. Add it to the password manager and lock up the printout.