Security considerations

- Storing secrets and the Nix store

Storing secrets and the Nix store

Nix stores every derivation that it builds in the Nix store. However, to make sure that everything in the Nix store is perfectly deterministic and usable by anyone, it needs to set the attributes of all files to a fixed value - this means that every file creation and modification date is set to UNIX timestamp `0`, but it *also* means that every file is made world-readable.

That's a problem when you're handling sensitive data!

For this reason, you should avoid storing any kind of secret or sensitive data in the Nix store, like passwords, credentials, API keys, SSH keys, and so on. Most NixOS modules will provide a mechanism for specifying secrets in some out-of-band way, usually by expecting you to specify the path to a 'key file' - a file somewhere outside of the Nix store, that is not managed by Nix, which contains the secret value(s). However, not all modules have been updated to do this yet, so for now you should always pay attention to where your secrets are being stored.

Why can't Nix just manage the secrets outside of the Nix store?

For the Nix model to work as designed, it needs to have a single universal index of build artifacts (store paths), and that is the Nix store. If secrets were managed outside of the store, Nix would not be able to provide all of its guarantees. That is why this task is usually left to tools that are dedicated to the purpose - for example, Morph can handle this for you for server deployments.