

ToiletPaper #145

Goooooooo, gopass!

Autor: Michael Braun / Project Manager / Platforms & Operations

✗ Problem

Die Verwaltung von Passwörtern in Projektteams ist meistens entweder mit Kosten verbunden (Cloud-Lösungen), funktioniert nur mäßig gut und ist oft chaotisch (KeePass-File, das irgendwo liegt) oder ist nur eingeschränkt nutzbar (nur intern erreichbare Passwortserver).

Zudem haben alle Passwortspeicher (inkl. dem hier im Nachgang vorgestellten) das Problem, dass selbst nachdem der Zugriff entzogen wurde, über eine lokale ältere Kopie weiterhin Zugriff auf alte Passwörter zugegriffen werden kann. Die einzige Lösung dafür bleibt leider, alle Passwörter zu rotieren.

✓ Lösung

Für alles andere gibt es: **gopass**! Hierbei handelt es sich um einen Passwortmanager, der alle Passwörter mit GPG verschlüsselt in git (sollte nicht public sein, da Metadaten nicht verschlüsselt sind) ablegt.

Durch die Kompatibilität zu [pass](#) gibt es ein Sammelsurium an Tools, Clients (Windows/Mac/Linux/Android/iOS) und Extensions für die meisten Browser. Über Wrapper-Skripte und fast beliebige Formatierung der Einträge kann man nochmal ganz viel Magie reinbringen und gopass sehr individuell nutzen. Es kann also von jedem Teammitglied gut bedient werden.

Weiterhin organisiert gopass Passwörter über (multi-)stores, wodurch Zugriffe sehr gut gesteuert werden können. Diese stores können dann von berechtigten Personen lokal eingebunden werden, um beispielhaft folgende Struktur zu schaffen:

Beispieldaten

```
$ gopass ls
gopass
jambit (/path/to/store/jambit)
jambit.com/
  mbraun
  project (/path/to/store/project)
    |aws.com/
    | project.mbraun
  personal (/path/to/store/personal)
    aws.com/
    personal.mbraun

$ gopass recipients
gopass
jambit (/path/to/store/jambit)
UID1 - M. Braun <m@jambit.com>
project (/path/to/store/project)
|UID1 - M. Braun <m@jambit.com>
|UID2 - K. Korn <kk@jambit.com>
UID3 - C. Grube <cg@jambit.com>
personal (/path/to/store/personal)
  UID1 - M. Braun <m@jambit.com>
  UID4 - M. Braun <m@gmail.com>
```

Nutzung

```
# show entry
$ gopass show jambit/jambit.com/mbraun
Secret: jambit/jambit.com/mbraun
THISISSTHESECRET!!!!
---
username: mbraun
azure-sub: 12345678-1234-5678-9987-
65432112
totp: Base32encodedsecret

# copy password (first line of entry) to
clipboard
$ gopass show -c jambit/jambit.com/mbraun
Copied jambit/jambit.com/mbraun to
clipboard.

# copy and display OTP token
$ gopass otp -c jambit/jambit.com/mbraun
561091 lasts 18s
Copied token for jambit/jambit.com
/mbraun to clipboard.

# copy key/value from entry
$ gopass show -c jambit/jambit.com/mbraun
azure-sub
Copied jambit/jambit.com/mbraun to
clipboard.
Secret: jambit/jambit.com/mbraun
Key: azure-sub
12345678-1234-5678-9987-65432112
```

⊕ Weiterführende Aspekte

- Cheat Sheet: <https://woile.github.io/gopass-cheat-sheet/>
- Quick Reference: <https://domaindrivenarchitecture.org/posts/2020-01-29-GopassQuickRef/>
- Gute Browser-Extension: <https://github.com/browserpass/browserpass-extension>