Pass Password Manager Cheatsheet

A comprehensive cheat sheet for Pass, the standard Unix password manager. Covers essential commands for password generation, storage, retrieval, and synchronization with Git.

Basic Usage

Initialization

Initialize Pass with a GPG ID:

```
pass init <gpg-id>
```

This command initializes the password store with the specified GPG ID for encryption.

Initialize Git for Password Store:

```
pass git init
```

This command initializes a Git repository within the password store for version control.

Add Remote Git Repository:

```
pass git remote add origin
<your.git:repository>
```

Adds a remote Git repository to synchronize passwords across multiple devices.

Storing Passwords

Insert a New Password:

```
pass insert [-m] <entry-name>
```

Inserts a new password entry. The `-m` option allows multiline input.

Generate a New Password:

```
pass generate [-n] <entry-name> <length>
```

Generates a new password of specified length. `-n` omits symbols.

Store Password with Custom Fields:

```
pass insert email.com/john
```

Then enter password in the first line, additional data below. Useful for storing usernames, security questions, etc.

Retrieving Passwords

List Password Entries:

```
pass ls [<path>]
```

Lists all password entries, optionally under the specified path.

Show Password Entry:

```
pass show <entry-name>
```

Shows the password for the specified entry and copies it to the clipboard.

Copy Password to Clipboard Without Showing:

```
pass -c <entry-name>
```

Copies the password to the clipboard without displaying it in the terminal.

Advanced Operations

Searching and Editing

Find Password Entries:

```
pass find <search-term>
```

Finds password entries matching the search term.

Edit Password Entry:

```
pass edit <entry-name>
```

Opens the password entry in a text editor to modify its content.

Moving, Copying, and Removing Entries

Move Password Entry:

```
pass mv <old-entry-name> <new-entry-
name>
```

Moves a password entry from the old name to the new name.

Copy Password Entry:

```
pass cp <source-entry-name> <dest-entry-
name>
```

Copies a password entry to a new entry name.

Remove Password Entry:

```
pass rm [-rf] <entry-name>
```

Removes a password entry. The `-r` option removes directories recursively, and `-f` forces removal without confirmation.

Synchronization with Git

Push Changes to Remote Repository:

```
pass git push
```

Pushes local changes to the remote Git repository.

Pull Changes from Remote Repository:

```
pass git pull
```

Pulls changes from the remote Git repository to the local password store.

Commit Changes

```
pass git commit -m "Your commit message"
```

Commit the changes to your local repository.

Configuration and Security

GPG Key Management

Change GPG ID:

```
pass init <new-gpg-id>
```

Changes the GPG ID used for encrypting the password store. This requires re-encrypting all passwords.

Add a Recipient:

```
pass add-recipient <gpg-id>
```

Adds a recipient to the password store, allowing them to decrypt the passwords.

Remove a Recipient:

```
pass rm-recipient <gpg-id>
```

Removes a recipient from the password store, revoking their ability to decrypt the passwords.

Password Generation Options

Customize Generated Passwords:

You can customize the characters used in generated passwords by modifying the `PASSWORD_CHARS` variable in the `~/.password-store/.gpg-id` file.

Exclude Symbols:

```
pass generate -n <entry-name> <length>
```

Generates a password without including symbols.

Security Best Practices

Regularly Synchronize:

Use `pass git push` and `pass git pull` frequently to keep your password store synchronized and backed up.

Secure GPG Key:

Protect your GPG private key with a strong passphrase and consider using a hardware security key.

Regularly Audit Passwords:

Periodically review and update your passwords to maintain strong security.

Integrating with Other Tools

Using with Browser Extensions

Browser Extensions:

Integrate Pass with browser extensions like `passff` (Firefox) or `chromeIPass` (Chrome) for seamless password retrieval and auto-filling in web browsers.

Install and Configure:

Install the browser extension and configure it to point to your password store directory (usually `~/.password-store`).

Scripting and Automation

Automate Password Retrieval:

Use `pass show` in scripts to retrieve passwords programmatically for automated tasks or system configurations. Ensure proper security measures are in place when using passwords in scripts.

Example Script (Bash):

```bash
#!/bin/bash
PASSWORD=$(pass show myapp/password)
echo "Password for myapp: $PASSWORD"
```

Troubleshooting

GPG Errors:

If you encounter GPG errors, ensure your GPG agent is running and properly configured. Check your `~/.gnupg/gpg.conf` and `~/.bashrc` files.

Git Synchronization Issues:

Resolve Git conflicts by merging changes or stashing local modifications before pulling. Use `pass git status` to check the status of your password store repository.