Using Mutt with Microsoft Cloud

If you're using CS email, this is not the right place to be. Get yourself configured using our Mail Client Guide.

If you would like to connect Mutt to Microsoft Office 365, keep reading. This tip is provided in the hopes that it will be useful. IT Services does not share in administration nor information and planning about their services with Techstaff, and we have no special access to their environment.

Using Modern Authentication

You **must** enable oath to authenticate to the O365. Support for password authentication is not likely to continue, or has already been removed. This guide was adapted using information from Mutt's OAUTH2 README. It is highly recommended that you read their own documents on this subject as well as this one.

What You Need

- Non-interactive GPG with your own keys.
- Mutt version 2.
- mutt_oauth2.py from mutt_oauth2.py saved in some place, and marked as executable.
- The Application ID that is also recorded in Microsoft's servers: "695da824-3d11-464b-98cc-83c714d21cf8".

Configuration

In .muttrc, set `imap_authenticators` and the path to the python script referenced above.

```
set imap_user = "YOUR_CNETID@uchicago.edu"
set imap_authenticators = "xoauth2"
set imap_oauth_refresh_command = "~/.mutt/mutt_oauth2.py
~/.mutt/o365.tokens"
```

Additionally, edit the `mutt_oauth2.py` script to include your GPG identity and the application ID above.

```
ENCRYPTION_PIPE = ['gpg', '--encrypt', '--recipient', 'YOUR_GPG_IDENTITY']
...
'client_id': '695da824-3d11-464b-98cc-83c714d21cf8',
```

Bootstrapping

The Modern Authentication protocol is essentially an application password that the python script will establish and keep updated inside the token file that is provided as an argument to it.

Mutt periodically executes the script while it is running in order to keep the password fresh. The token file is encrypted on disk and decrypted by the python script when it runs. The file is updated when the server indicates that the password has changed. In this way, the scheme forces users to setup a complex password and use a password manager without their knowledge. The token file can be moved between computers, and you can refresh it using the python script (Mutt is not involved).

To get your new credential into the token file, execute the script **once** with the argument `–authorize`, and a path to a (not-yet-created) token file.

```
~/.mutt/mutt_oauth2.py --authorize --verbose --authflow authcode
~/.mutt/o364.tokens
```

The script will prompt you with questions and directions. Paste the URL into the browser and complete the authentication. For your author, the token then had to be extracted from the resulting URL parameters. Yuck.

If you leave your Mutt disconnected for a while, you will have to bootstrap again. After a few months (unclear), you will also have to do this again anyway.

Running

After you have bootstrapped, you can continue to use Mutt as normal.

Tips

- After using the browser to authenticate during bootstrapping, you might see a single white page or nothing at all. Check the address bar for the long secret parameter that the script is waiting for you to paste.
- After some time (month or more), you will have to bootstrap again. If the token is allowed to expire without refreshing it, you will also have to authorize from the start.
- Microsoft does not intend to restrict SMTP Authentication, but testing shows that Modern Authentication is implemented on port 587. You can use this capability by changing .muttrc

```
set smtp_authenticators = xoauth2
set smtp_url = "smtp://${imap_user}@smtp.office365.com:587/"
```