mpop

A POP3 client
version 1.4.21, 12 December 2024

Martin Lambers (`marlam@marlam.de`)

This manual was last updated 12 December 2024 for version 1.4.21 of mpop.

Table of Contents

1 Introduction

mpop is a POP3 client.

In its default mode of operation, it retrieves mails from one or more POP3 mailboxes, optionally does some filtering, and delivers them through a mail delivery agent (MDA), to a maildir folder, or to an mbox file. Mails that were successfully delivered before will not be retrieved a second time, even if errors occur or mpop is terminated in the middle of a session.

The best way to start is probably to have a look at the Examples section. See Chapter 9 [Examples], page 20.

In addition to the mail retrieval mode, mpop can be used in server information mode. In this mode, mpop prints as much information as it can get about a given POP3 server (greeting, supported features, login delay, maximum mail size, ...).

Normally, a configuration file contains information about which POP3 server to use and how to use it, but all settings can also be configured on the command line.

POP3 server information is organized in accounts. Each account describes one POP3 server: host name, authentication settings, TLS settings, and so on. Each configuration file can define multiple accounts.

Supported features include:

- Header based mail filtering: filter junk mail before downloading it
- Delivery to maildir folders, mbox files, Exchange pickup directories, or a mail delivery agent (MDA)
- Very fast POP3 implementation, using command pipelining
- TLS secured connections (including server certificate verification and the possibility to send a client certificate)
- Authentication methods USER/PASS, APOP, PLAIN, LOGIN and CRAM-MD5 (and GSSAPI, SCRAM-SHA-1, SCRAM-SHA-256, DIGEST-MD5, and NTLM when GNU SASL is used)
- Internationalized Domain Names (IDN)

2 Configuration file

A suggestion for a suitable configuration file can be generated using the '`--configure`' option; see [–configure], page 9. The default configuration file is `~/.mpoprc` or `$XDG_CONFIG_HOME/mpop/config`. Settings in this file can be changed by command line options.

A configuration file is a simple text file. Empty lines and comment lines (first non-blank character is '#') are ignored. Every other line must contain a command and may contain an argument to that command. The argument may be enclosed in double quotes (").

If a file name starts with the tilde (`~`), this tilde will be replaced by `$HOME`.

If a command accepts the argument '`on`', it also accepts an empty argument and treats that as if it was '`on`'.

Commands are organized in accounts. Each account starts with the '`account`' command and defines the settings for one POP3 account.

See Chapter 9 [Examples], page 20.

2.1 General commands

'`defaults`'
: Set defaults. The following commands will set default values for all following account definitions.

'`account` *name* `[ :` *account*`[,...] ]`'
: Start a new account definition with the given name. The current default values are filled in (see [defaults], page 2).
If a colon and a list of previously defined accounts is given after the account name, the new account, with the filled in default values, will inherit all settings from the accounts in the list.

'`eval [`*cmd*`]`'
: Replace the current configuration file line with the first line of the output (stdout) of the command *cmd*. This can be used to decrypt settings or to create them via scripts. For example, `eval echo host localhost` replaces the current line with `host localhost`.
Note that every '`eval`' line will be evaluated when the configuration file is read.
Note that for passwords you can also use the [passwordeval], page 3, command instead of `eval password cmd`. This has the advantage that the command is only evaluated if needed.

'`host` *hostname*'
: The POP3 server to retrieve mails from. The argument may be a host name or a network address. Every account definition must contain this command.

'`port` *number*'
: The port that the POP3 server listens on. The default is 110 ("pop3"), unless TLS without STARTTLS is used, in which case it is 995 ("pop3s").

'`source_ip [`*IP*`]`'
: Set a source IP address to bind the outgoing connection to. Useful only in special cases on multi-home systems. An empty argument disables this.

‘`proxy_host [IP|hostname]`’
: Use a SOCKS proxy. All network traffic will go through this proxy host, including DNS queries, except for a DNS query that might be necessary to resolve the proxy host name itself (this can be avoided by using an IP address as proxy host name). An empty argument disables proxy usage. The supported SOCKS protocol version is 5. If you plan to use this with Tor, see also Section 9.3 [Using mpop with Tor], page 22.

‘`proxy_port [number]`’
: Set the port number for the proxy host. An empty ‘`number`’ argument resets this to the default port, which is 1080 (`"socks"`).

‘`socket [socketname]`’
: Set the file name of a unix domain socket to connect to. This overrides both ‘`host`’/‘`port`’ and ‘`proxy_host`’/‘`proxy_port`’.

‘`timeout (off|seconds)`’
: Set or unset a network timeout, in seconds. The default is 180 seconds. The argument ‘`off`’ means that no timeout will be set, which means that the operating system default will be used.

‘`pipelining (auto|on|off)`’
: Enable or disable POP3 pipelining. You should never need to change the default setting, which is ‘`auto`’: mpop enables pipelining for POP3 servers that advertise this capability, and disables it for all other servers. Pipelining can speed up a POP3 session substantially.

2.2 Authentication commands

See Chapter 5 [Authentication], page 14.

‘`auth [(on|method)]`’
: Choose an authentication method. The default argument ‘`on`’ chooses a method automatically. Accepted methods are ‘`scram-sha-256-plus`’, ‘`scram-sha-1-plus`’, ‘`scram-sha-256`’, ‘`scram-sha-1`’, ‘`user`’, ‘`apop`’, ‘`plain`’, ‘`gssapi`’, ‘`external`’, ‘`oauthbearer`’, ‘`cram-md5`’, ‘`digest-md5`’, ‘`login`’, ‘`ntlm`’, and ‘`xoauth2`’.

‘`user [username]`’
: Set the user name for authentication. An empty argument unsets the user name.

‘`password [secret]`’
: Set the password for authentication. An empty argument unsets the password. Consider using the ‘`passwordeval`’ command or a key ring instead of this command, to avoid storing cleartext passwords in the configuration file.

‘`passwordeval [cmd]`’
: Set the password for authentication to the output (stdout) of the command *cmd*. This can be used e.g. to decrypt password files on the fly or to query key rings, and thus to avoid storing cleartext passwords.

‘`ntlmdomain [ntlmdomain]`’
: Set a domain for the ‘`ntlm`’ authentication method. This is obsolete.

2.3 TLS commands

See Chapter 4 [Transport Layer Security], page 12.

‘`tls [(on|off)]`’
: Enable or disable TLS (also known as SSL) for secured connections.

‘`tls_starttls [(on|off)]`’
: Choose the TLS variant: start TLS from within the session (‘`on`’, default), or tunnel the session through TLS (‘`off`’).

‘`tls_trust_file [file]`’
: Activate server certificate verification using a list of trusted Certification Authorities (CAs). The default is the special value ‘`system`’, which selects the system default. An empty argument disables trust in CAs. If you select a file, it must be in PEM format, and you should also use ‘`tls_crl_file`’.

‘`tls_crl_file [file]`’
: This sets a certificate revocation list (CRL) file for TLS, to check for revoked certificates (an empty argument, which is the default, disables this).
OCSP is an alternative to CRL files. When GnuTLS is used, stapled OCSP information will be checked automatically, and the MustStaple TLS extension is supported, however no manual OCSP queries will be sent when stapled OCSP information is missing. With other TLS libraries, behavior may be different.

‘`tls_fingerprint [fingerprint]`’
: Set the fingerprint of a single certificate to accept for TLS. This certificate will be trusted regardless of its contents (this overrides ‘`tls_trust_file`’). The fingerprint should be of type SHA256, but can for backwards compatibility also be of type SHA1 or MD5 (please avoid this). The format should be `01:23:45:67:...`. Use ‘`--serverinfo --tls --tls-certcheck=off --tls-fingerprint=`’ to get the server certificate fingerprint.

‘`tls_key_file [file]`’
: Send a client certificate to the server (use this together with ‘`tls_cert_file`’). The file must contain the private key of a certificate in PEM format. An empty argument disables this feature.

‘`tls_cert_file [file]`’
: Send a client certificate to the server (use this together with ‘`tls_key_file`’). The file must contain a certificate in PEM format. An empty argument disables this feature.

‘`tls_certcheck [(on|off)]`’
: Enable or disable checks of the server certificate. They are enabled by default. Disabling them will override ‘`tls_trust_file`’ and ‘`tls_fingerprint`’. WARNING: When the checks are disabled, TLS sessions will not be secure!

‘`tls_priorities [priorities]`’
: Set priorities for TLS session parameters. The default is set by the TLS library and can be selected by using an empty argument to this command. The interpretation of the *priorities* string depends on the TLS library. Use ‘`--version`’ to find out which TLS library you use.
For GnuTLS, see the section on Priority Strings in the manual.
For libtls, the *priorites* string is a space-separated list of parameter strings prefixed with either `PROTOCOLS=`, `CIPHERS=`, or `ECDHECURVES=`. These parameter strings will be passed to the functions ‘`tls_config_parse_protocols`’, ‘`tls_config_set_ciphers`’, and ‘`tls_config_set_ecdhecurves`’. Unrecognized parts of the *priorities* string will be ignored. Example: `PROTOCOLS=TLSv1.3 CIPHERS=ECDHE-RSA-AES128-SHA256 ECDHECURVES=P-384`.

‘`tls_host_override [host]`’
: By default, TLS host verification uses the host name given by the ‘`host`’ command. This command allows one to use a different host name for verification. This is only useful in special cases.

‘`tls_min_dh_prime_bits [bits]`’
: Deprecated, use ‘`tls_priorities`’ instead. Set or unset the minimum number of Diffie-Hellman (DH) prime bits accepted for TLS sessions. The default is set by the TLS library and can be selected by using an empty argument to this command. Only lower the default (for example to 512 bits) if there is no other way to make TLS work with the remote server.

2.4 Commands specific to mail retrieval mode

See Chapter 6 [Mail retrieval mode], page 16.

‘`delivery method method_arguments...`’
: How to deliver messages received from this account.

- delivery mda *command*
  Deliver the mails through a mail delivery agent (MDA).
  All occurrences of `%F` in the command will be replaced with the envelope from address of the current message (or MAILER-DAEMON if none is found). Note that this address is guaranteed to contain only letters `a-z` and `A-Z`, digits `0-9`, and any of `.@_-+/`, even though that is only a subset of what is theoretically allowed in a mail address. Other characters, including those interpreted by the shell, are replaced with `_`. Nevertheless, you should put `%F` into single quotes: `'%F'`.
  Use `delivery mda "/usr/bin/procmail -f '%F' -d $USER"` for the procmail MDA.
  Use `delivery mda "/usr/sbin/sendmail -oi -oem -f '%F' -- $USER"` to let your MTA handle the mail.
  Use `delivery mda /usr/local/bin/msmtp --host=localhost --from='%F' -- $USER@`hostname`.`dnsdomainname`"` to pass the mail to your MTA via SMTP.

- delivery maildir *directory*
  Deliver the mails to the given maildir directory. The directory must exist and it must have the maildir subdirectories '`cur`', '`new`', and '`tmp`'; mpop will not create directories. This delivery type only works on file systems that support hard links.
- delivery mbox *mbox-file*
  Deliver the mails to the given file in mbox format. The file will be locked with `fcntl(2)`. mpop uses the MBOXRD mbox format variant; see the documentation of the mbox format.
- delivery exchange *directory*
  Deliver the mails to the given Exchange pickup directory. The directory must exist.

If the delivery method needs to parse the mail headers for an envelope from address (the mda method if the command contains `%F`, and the mbox method), then it needs to create a temporary file to store the mail headers (but not the body).

'`uidls_file` *`filename`*'
: The file to store UIDLs in. These are needed to identify new messages. `%U` in the filename will be replaced by the username of the current account. `%H` in the filename will be replaced by the hostname of the current account. If the filename contains directories that do not exist, mpop will create them. mpop locks this file for exclusive access when accessing the associated POP3 account. The default value is `~/.mpop_uidls/%U_at_%H`. You can also use a single UIDLS file for multiple accounts, but then you cannot poll more than one of these accounts at the same time.

'`only_new [(on|off)]`'
: By default, mpop processes only new messages (new messages are those that were not already successfully retrieved in an earlier session). If this option is turned off, mpop will process all messages.

'`keep [(on|off)]`'
: Keep all mails on the POP3 server, never delete them. The default behavior is to delete mails that have been successfully delivered or filtered by kill filters.

'`killsize (off|`*`size`*`)`'
: Mails larger than the given size will be deleted, not downloaded (unless the keep command is used, in which case they will just be skipped). The size argument must be zero or greater. If it is followed by a 'k' or an 'm', the size is measured in kibibytes/mebibytes instead of bytes. Note that some POP3 servers report slightly incorrect sizes for mails. See Chapter 8 [Filtering], page 19.
  When '`killsize`' is set to 0 and '`keep`' is set to on, then all mails are marked as retrieved, but no mail gets deleted from the server. This can be used to synchronize the UID list on the client to the UID list on the server.

'`skipsize (off|`*`size`*`)`'
: Mails larger than the given size will be skipped (not downloaded). The size argument must be zero or greater. If it is followed by a 'k' or an 'm', the size is mea-

sured in kibibytes/mebibytes instead of bytes. Note that some POP3 servers report slightly incorrect sizes for mails. See Chapter 8 [Filtering], page 19.

‘`filter [COMMAND]`’

Set a filter which will decide whether to retrieve, skip, or delete each mail by investigating the mail's headers. The POP3 server must support the POP3 TOP command for this to work; see Chapter 7 [Server information mode], page 17. An empty argument disables filtering.

All occurrences of `%F` in the command will be replaced with the envelope from address of the current message (or MAILER-DAEMON if none is found). Note that this address is guaranteed to contain only letters `a-z` and `A-Z`, digits `0-9`, and any of `.@_-+/`, even though that is only a subset of what is theoretically allowed in a mail address. Other characters, including those interpreted by the shell, are replaced with `_`. Nevertheless, you should put `%F` into single quotes: `'%F'`.

All occurrences of `%S` in the command will be replaced with the size of the current mail as reported by the POP3 server.

The mail headers (plus the blank line separating the headers from the body) will be piped to the command. Based on the return code, mpop decides what to do with the mail:

- 0: proceed normally; no special action
- 1: delete the mail; do not retrieve it
- 2: skip the mail; do not retrieve it

Return codes greater than or equal to 3 mean that an error occurred. The `sysexits.h` error codes may be used to give information about the kind of the error, but this is not necessary. See Chapter 8 [Filtering], page 19.

‘`received_header [(on|off)]`’

Enable or disable adding a Received header. By default, mpop prepends a Received header to the mail during delivery. This is required by the RFCs if the mail is subsequently further delivered e.g. via SMTP.

3 Invocation

3.1 Synopsis

- Mail retrieval mode (default):
  `mpop [`*`option`*`...] [--] [`*`account`*`...]`

- Configuration mode:
  `mpop --configure` *`mailaddress`*

- Server information mode:
  `mpop [`*`option`*`...] --serverinfo [`*`account`*`...]`

mpop is usually run with one or more accounts as parameters. If no account is provided, an account named '`default`' is used if it exist. Alternatively, `mpop -a` will use all accounts defined in the configuration file.

This can be automated by running mpop from `cron(8)`.

3.2 Exit code

The standard exit codes from `sysexits.h` are used.

3.3 Files

'`~/.mpoprc or $XDG_CONFIG_HOME/mpop/config.`'
: The default user configuration file.

'`~/.mpop_uidls`'
: Default directory to store UIDLs files in.

'`~/.netrc and SYSCONFDIR/netrc`'
: The `netrc` file contains login information. Before prompting for a password, msmtp will search it in ~/.netrc and SYSCONFDIR/netrc.

3.4 Environment

'`$USER, $LOGNAME`'
: These variables override the user's login name. `$LOGNAME` is only used if `$USER` is unset. The user's login name is used for `Received` headers.

3.5 Options

Options override configuration file settings. The following options are accepted:

3.5.1 General options

'`--version`'
: Print version information, including information about the libraries used.

‘`--help`’ Print help.

‘`-P`’
‘`--pretend`’
Print the configuration settings that would be used, but do not take further action. An asterisk (’*’) will be printed instead of the password.

‘`-d`’
‘`--debug`’ Print lots of debugging information, including the whole conversation with the server. Be careful with this option: the (potentially dangerous) output will not be sanitized, and your password may get printed in an easily decodable format! This option implies `--half-quiet`, because the debugging output would otherwise interfere with the progress output.

3.5.2 Changing the mode of operation

‘`--configure=`*`mailaddress`*’
Generate a configuration for the given mail address and print it. This can be modified or copied unchanged to the configuration file. Note that this only works for mail domains that publish appropriate SRV records; see RFC 8314.

‘`-S`’
‘`--serverinfo`’
Print information about the POP3 server and exit. This includes information about supported features (pipelining, authentication methods, TOP command, . . .), about parameters (time for which mails will not be deleted, minimum time between logins, . . .), and about the TLS certificate (if TLS is active). See Chapter 7 [Server information mode], page 17.

3.5.3 Configuration options

Most options in this category correspond to a configuration file command. Please refer to Chapter 2 [Configuration file], page 2, for detailed information.

‘`-C` *`filename`*’
‘`--file=`*`filename`*’
Use the given file instead of `~/.mpoprc` or `XDG_CONFIG_HOME/mpop/config` as the configuration file.

‘`--host=`*`hostname`*’
Use this server with settings from the command line; do not use any configuration file data. This option disables loading of the configuration file. You cannot use both this option and account names on the command line.

‘`--port=`*`number`*’
Set the port number. See [port], page 2.

‘`--source-ip=[`*`IP`*`]`’
Set or unset an IP address to bind the socket to. See [source_ip], page 2.

‘`--proxy-host=[`*`IP`*`|`*`hostname`*`]`’
Set or unset a SOCKS proxy to use. See [proxy_host], page 2.

‘`--proxy-port=[number]`’
: Set or unset a port number for the proxy host. See [proxy_port], page 3.

‘`--socket=[socketname]`’
: Set or unset a local unix domain socket name to connect to. See [socket], page 3.

‘`--timeout=(off|seconds)`’
: Set or unset a network timeout, in seconds. See [timeout], page 3.

‘`--pipelining=(auto|on|off)`’
: Enable or disable POP3 pipelining. See [pipelining], page 3.

‘`--received-header[=(on|off)]`’
: Enable or disable the Received header. See [received_header], page 7.

‘`--auth[=(on|method)]`’
: Set the authentication method to automatic (with ‘`on`’) or manually choose an authentication method. See [auth], page 3.

‘`--user=[username]`’
: Set or unset the user name for authentication. See [user], page 3.

‘`--passwordeval=[eval]`’
: Evaluate password for authentication. See [passwordeval], page 3.

‘`--tls[=(on|off)]`’
: Enable or disable TLS/SSL. See [tls], page 4.

‘`--tls-starttls[=(on|off)]`’
: Enable or disable STARTTLS for TLS. See [tls_starttls], page 4.

‘`--tls-trust-file=[file]`’
: Set or unset a trust file for TLS. See [tls_trust_file], page 4.

‘`--tls-crl-file=[file]`’
: Deprecated. Set or unset a certificate revocation list (CRL) file for TLS. See [tls_crl_file], page 4.

‘`--tls-fingerprint=[fingerprint]`’
: Set ot unset the fingerprint of a trusted TLS certificate. See [tls_fingerprint], page 4.

‘`--tls-key-file=[file]`’
: Set or unset a key file for TLS. See [tls_key_file], page 4.

‘`--tls-cert-file=[file]`’
: Set or unset a cert file for TLS. See [tls_cert_file], page 4.

‘`--tls-certcheck[=(on|off)]`’
: Enable or disable server certificate checks for TLS. See [tls_certcheck], page 4.

‘`--tls-priorities=[priorities]`’
: Set or unset TLS priorities. See [tls_priorities], page 4.

‘`--tls-host-override=[host]`’
: Set or unset override for TLS host verification. See [tls_host_override], page 5.

‘`--tls-min-dh-prime-bits=[bits]`’
: Deprecated, use ‘`--tls-priorities`’ instead. Set or unset minimum bit size of the Diffie-Hellman (DH) prime. See [tls_min_dh_prime_bits], page 5.

3.5.4 Options specific to mail retrieval mode

‘`-q`’
‘`--quiet`’ Do not print status or progress information.

‘`-Q`’
‘`--half-quiet`’
: Print status but not progress information.

‘`-a`’
‘`--all-accounts`’
: Query all accounts in the configuration file.

‘`-A`’
‘`--auth-only`’
: Authenticate only; do not retrieve mail. Useful for SMTP-after-POP.

‘`-s`’
‘`--status-only`’
: Print number and size of mails in each account only; do not retrieve mail.

‘`-n`’
‘`--only-new[=(on|off)]`’
: Process only new messages. See [only_new], page 6.

‘`-k`’
‘`--keep[=(on|off)]`’
: Do not delete mails from POP3 servers, regardless of other options or settings. See [keep], page 6.

‘`--killsize=(off|size)`’
: Set or unset kill size. See [killsize], page 6.

‘`--skipsize=(off|size)`’
: Set or unset skip size. See [skipsize], page 6.

‘`--filter=[command]`’
: Set a filter which will decide whether to retrieve, skip, or delete each mail by investigating the mail’s headers. See [filter], page 7.

‘`--delivery=method,method_arguments...`’
: How to deliver messages received from this account. See [delivery], page 5. Note that a comma is used instead of a blank to separate the method from its arguments.

‘`--uidls-file=filename`’
: File to store UIDLs in. See [uidls_file], page 6.

4 Transport Layer Security

Transport Layer Security (TLS) "... provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery" (quote from RFC2246).

A server can use TLS in one of two modes:

- Via a STARTTLS command
  The session starts with the normal protocol initialization, and TLS is then started using the protocol's STARTTLS command.
- Immediately
  TLS is initialized before the normal protocol initialization. This requires a separate port.

The first mode is the default, but you can switch to the second mode by disabling [tls_starttls], page 4.

When TLS is started, the server sends a certificate to identify itself. To verify the server identity, a client program is expected to check that the certificate is formally correct and that it was issued by a Certificate Authority (CA) that the user trusts. (There can also be certificate chains with intermediate CAs.)

The list of trusted CAs is specified using the [tls_trust_file], page 4, command. The default value ist '`system`' and chooses the system-wide default, but you can also choose the trusted CAs yourself.

A fundamental problem with this is that you need to trust CAs. Like any other organization, a CA can be incompetent, malicious, subverted by bad people, or forced by government agencies to compromise end users without telling them. All of these things happened and continue to happen worldwide. The idea to have central organizations that have to be trusted for your communication to be secure is fundamentally broken.

Instead of putting trust in a CA, you can choose to trust only a single certificate for the server you want to connect to. For that purpose, specify the certificate fingerprint with [tls_fingerprint], page 4. This makes sure that no man-in-the-middle can fake the identity of the server by presenting you a fraudulent certificate issued by some CA that happens to be in your trust list. However, you have to update the fingerprint whenever the server certificate changes, and you have to make sure that the change is legitimate each time, e.g. when the old certificate expired. This is inconvenient, but it's the price to pay.

Information about a server certificate can be obtained with '`--serverinfo --tls --tls-certcheck=off`'. This includes the issuer CA of the certificate (so you can trust that CA via '`tls_trust_file`'), and the fingerprint of the certificate (so you can trust that particular certificate via '`tls_fingerprint`'). See Chapter 7 [Server information mode], page 17.

If you need to fine tune TLS parameters, have a look at the [tls_priorities], page 4, command.

4.1 Client Certificates

TLS also allows the server to verify the identity of the client. For this purpose, the client has to present a certificate issued by a CA that the server trusts. To present that certificate, the client also needs the matching key file. You can set the certificate and key files using [tls_cert_file], page 4, and [tls_key_file], page 4. This mechanism can also be used to authenticate users, so that traditional user / password authentication is not necessary anymore. See the EXTERNAL mechanism in Chapter 5 [Authentication], page 14.

```
# Enable TLS
tls on
# Enable TLS client certificates
tls_cert_file /path/to/client_cert
tls_key_file /path/to/client_key
# Enable authentication via the EXTERNAL mechanism (optional; depends on server)
# The user name is empty because the server should get it from the client cert
auth external
user ""
```

You can also use client certificates stored on some external authentication device by specifying GnuTLS device URIs in [tls_cert_file], page 4, and [tls_key_file], page 4. You can find the correct URIs using `p11tool --list-privkeys --login` (p11tool is bundled with GnuTLS). If your device requires a PIN to access the data, you can specify that using one of the password mechanisms (e.g. [passwordeval], page 3, [password], page 3).

```
tls_cert_file pkcs11:model=PKCS%2315%20emulated;manufacturer=piv_II;serial=00000000;to
tls_key_file  pkcs11:model=PKCS%2315%20emulated;manufacturer=piv_II;serial=00000000;to
passwordeval  gpg2 --no-tty -q -d ~/.smart-card-pin.gpg
```

5 Authentication

POP3 servers require a client to authenticate before retrieving mail.

Usually a user name and a password are used for authentication. The user name specified in the configuration file with the [user], page 3, command. There are five different methods to specify the password:

1. Add the password to the system key ring.
   Currently supported key rings are the Gnome key ring and the Mac OS X Keychain. For the Gnome key ring, use the command '`secret-tool`' (part of Gnome's libsecret) to store passwords:

   ```
   $ secret-tool store --label=mpop \
     host pop.freemail.example \
     service pop3 \
     user joe.smith
   ```

   On Mac OS X, use the following command:

   ```
   security add-internet-password -s pop.freemail.example -r pop3 -a joe.smith -w
   ```

   In both examples, replace pop.freemail.example with the POP3 server name, and joe.smith with your user name.
2. Store the password in an encrypted files, and use [passwordeval], page 3, to specify a command to decrypt that file, e.g. using GnuPG. See Chapter 9 [Examples], page 20.
3. Store the password in the configuration file using the [password], page 3, command. (Usually it is not considered a good idea to store passwords in cleartext files. If you do it anyway, you must make sure that the file can only be read by yourself.)
4. Store the password in `~/.netrc`. This method is probably obsolete.
5. Type the password into the terminal when it is required.

It is recommended to use method 1 or 2.

Multiple authentication methods exist. Most servers support only some of them.

The following user / password methods are supported:

- '`SCRAM-SHA-1`' and '`SCRAM-SHA-1-PLUS`'
  A method that avoids cleartext passwords and requires the server to prove that it is in posession of the (hashed and salted) password, which prevents some man-in-the-middle-attacks. The '`-PLUS`' variant additionally uses TLS channel binding information for even better security guarantees.
- '`SCRAM-SHA-256`' and '`SCRAM-SHA-256-PLUS`'
  Same as the SCRAM-SHA-1 methods, but with a stronger hash function.
- '`USER`'
  A simple cleartext method supported by all servers.
- '`PLAIN`'
  Another simple cleartext method supported by almost all servers.
- '`APOP`'
  An obsolete method that avoids cleartext passwords, but is not considered secure anymore.

- ‘CRAM-MD5’
  An obsolete method that avoids cleartext passwords, but is not considered secure anymore.
- ‘DIGEST-MD5’
  An overcomplicated obsolete method that avoids cleartext passwords, but is not considered secure anymore.
- ‘LOGIN’
  A non-standard cleartext method similar to (but worse than) PLAIN.
- ‘NTLM’
  An obscure non-standard method that is now considered broken. It sometimes requires a special domain parameter passed via [ntlmdomain], page 3. Do not use it.

If no method is specified, mpop will autoselect one based on security benefits. With TLS, the order is ‘SCRAM-SHA-256-PLUS’, ‘SCRAM-SHA-1-PLUS’, ‘SCRAM-SHA-256’, ‘SCRAM-SHA-1’, ‘PLAIN’, ‘APOP’, ‘USER’, followed by some of the obsolete methods if nothing else is available. Without TLS, only ‘SCRAM-SHA-256’ and ‘SCRAM-SHA-1’ are considered.

There are currently three authentication methods that are not based on user / password information and have to be chosen manually:

- ‘OAUTHBEARER’ or its predecessor ‘XOAUTH2’
  An OAuth2 token from the mail provider is used as the password. See the documentation of your mail provider for details on how to get this token. The ‘passwordeval’ command can be used to pass the regularly changing tokens into mpop from a script or an environment variable.
- ‘EXTERNAL’
  The authentication happens outside of the protocol, typically by sending a TLS client certificate (see [Client Certificates], page 12).
  The EXTERNAL method merely confirms that this authentication succeeded; it does not perform the authentication. Thus it may not be necessary to use it for authentication to succeed, and if the server does not support the EXTERNAL method, this does not mean that it does not support authentication with TLS client certificates.
- ‘GSSAPI’
  With this method, the Kerberos framework takes care of secure authentication. Only a user name is required.

It depends on the underlying authentication library and its version whether a particular method is supported or not. Use `--version` to find out which methods are supported by your version.

6 Mail retrieval mode

In this mode, mpop retrieves mail from one or more POP3 servers. It delivers each of them using the method that was given with the [delivery], page 5, command or [–delivery], page 11, option.

While retrieving the mail, mpop displays approximate progress information, which can be turned off with the [–half-quiet], page 11, or [–quiet], page 11, options.

If the delivery succeeded, the mail is deleted from the POP3 server by default. The [keep], page 6, command and [–keep], page 11, option prevent the deletion of mails. Some POP3 servers will delete mails without any user interaction. See EXPIRE in Chapter 7 [Server information mode], page 17. Mpop can do nothing about that.

If you do not want to download certain mails, but skip them or delete them directly, you can do filtering based on the mail headers. See Chapter 8 [Filtering], page 19.

If you just want to know if you have new mails (and how many), use the [–status-only], page 11, option.

If you just want to authenticate to the POP3 server, but do not want to look at your mails, use the [–auth-only], page 11, option. This can be useful for sending mail through SMTP servers that require SMTP-after-POP (aka POP-before-SMTP).

Before mpop delivers a mail, it prepends a Received header to it. This is necessary for example if the delivery method transmits the mail to an SMTP server, but can be disabled with the [received_header], page 7, command. Mpop does not change the contents of the mail in any other way.

7 Server information mode

In server information mode, mpop prints as much information about the POP3 server as it can get and then exits.

The POP3 features that can be detected are:

- IMPLEMENTATION
  The implementation string of the POP3 server.
- CAPA
  Support for the POP3 CAPA command. The server sends a list of its capabilities in response to this command.
- PIPELINING
  Support for POP3 pipelining. See [pipelining], page 3.
- TOP
  Support for the POP3 TOP command. This is needed for header based filtering to work. See Chapter 8 [Filtering], page 19.
- UIDL
  Support for the POP3 UIDL command. This is needed to distinguish between new and already retrieved messages.
- LOGIN-DELAY
  The minimum time between two POP3 sessions. The server may refuse a POP3 session if the last one was active less than this time period ago.
- EXPIRE
  The time after which old mails are deleted by the POP3 server.
  - NEVER: The POP3 server will not delete mail without the user requesting it.
  - 0: The POP3 server will not keep mails; all mails will be deleted after they have been downloaded, regardless of the user's wishes.
  - *number*: The number of days that the POP3 server will keep mails before deleting them without user interaction.
- STARTTLS
  See Chapter 4 [Transport Layer Security], page 12.
- AUTH
  See Chapter 5 [Authentication], page 14.
- RESP-CODES
  If authentication fails and the POP3 server issues an error message beginning with a square bracket, this message will include additional information about the source of the error:
  - `[LOGIN-DELAY]`: The login delay period hast not yet expired.
  - `[IN-USE]`: Authentication succeeded but the mailbox is currently in use, possibly by another POP3 session.
- AUTH-RESP-CODE
  If authentication fails and the POP3 server issues an error message beginning with a square bracket, this message will include additional information about the source of the error:
  - `[LOGIN-DELAY]`: The login delay period hast not yet expired.

    - `[IN-USE]`: Authentication succeeded but the mailbox is currently in use, possibly by another POP3 session.
    - `[SYS/TEMP]`: Temporary system failure; try again later.
    - `[SYS/PERM]`: Permanent system failure; ask the administrator.
    - `[AUTH]`: Incorrect user name or password or some other problem with the user's credentials.

If TLS is activated for server information mode, the following information will be printed about the POP3 server's TLS certificate (if available):

- Owner information
    - Common Name
    - Organization
    - Organizational unit
    - Locality
    - State or Province
    - Country
- Issuer information
    - Common Name
    - Organization
    - Organizational unit
    - Locality
    - State or Province
    - Country
- General
    - Activation time
    - Expiration time
    - SHA256 fingerprint
    - SHA1 fingerprint (deprecated)

8 Filtering

There are three filtering commands available. They will be executed in the following order:

1. '`killsize`'
2. '`skipsize`'
3. '`filter`'

If a filtering command applies to a mail, the remaining filters will not be executed.

The POP3 server must support the POP3 TOP command (Chapter 7 [Server information mode], page 17) for filtering with a filter command: It is used to read the mail headers (plus the blank line separating the header from the body) and pipe them to the filter command.

Note that, if the filter decides that the mail should be retrieved, the complete mail has to be downloaded, including the headers, so the headers will be downloaded twice. This is because there's no way in POP3 to download just the mail body. Sometimes this overhead surpasses the savings of the filtering.

The filter command looks at the mail headers and signals with its exit code what mpop should do with the mail:

- 0: retrieve the mail
- 1: delete the mail; do not retrieve it
- 2: skip the mail; do not retrieve it

Return codes greater than or equal to 3 mean that an error occurred. The `sysexits.h` error codes may be used to give information about the kind of the error, but this is optional.

Since the filter command will be passed to a shell, you can use all shell command constructs in addition to just calling a script or program. This allows flexible filter constructs. See Section 9.2 [Filtering with SpamAssassin], page 22.

Some POP3 servers count end-of-line characters as two bytes (CRLF) instead of one (LF), so that the size of a mail as reported by the POP3 server is slightly larger than the actual size. The filters use the size values reported by the POP3 server since they cannot know the actual size in advance. Thus you cannot rely on *exact* size filtering.

9 Examples

9.1 A configuration file

```
# Example for a user configuration file ~/.mpoprc
#
# This file focusses on TLS, authentication, and the mail delivery method.
# Features not used here include mail filtering, timeouts, SOCKS proxies,
# TLS parameters, and more.


# Set default values for all following accounts.
defaults

# Always use TLS.
tls on

# Set a list of trusted CAs for TLS. The default is to use system settings, but
# you can select your own file.
#tls_trust_file /etc/ssl/certs/ca-certificates.crt

# Deliver mail to an MBOX mail file:
delivery mbox ~/Mail/inbox
# Deliver mail to a maildir folder:
#delivery maildir ~/Mail/incoming
# Deliver mail via procmail:
#delivery mda "/usr/bin/procmail -f '%F' -d $USER"
# Deliver mail via the local SMTP server:
#delivery mda "/usr/bin/msmtp --host=localhost --from='%F' -- $USER"
# Deliver mail to an Exchange pickup directory:
#delivery exchange c:\exchange\pickup

# Use an UIDLS file in ~/.local/share instead of ~/.mpop_uidls
uidls_file ~/.local/share/%U_at_%H


# A freemail service
account freemail

# Host name of the POP3 server
host pop.freemail.example

# As an alternative to tls_trust_file, you can use tls_fingerprint
# to pin a single certificate. You have to update the fingerprint when the
# server certificate changes, but an attacker cannot trick you into accepting
# a fraudulent certificate. Get the fingerprint with
```

```
# $ mpop --serverinfo --tls --tls-certcheck=off --host=pop.freemail.example
#tls_fingerprint 00:11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:00:11:22:33

# Authentication. The password is given using one of five methods, see below.
user joe.smith

# Password method 1: Add the password to the system keyring, and let mpop get
# it automatically. To set the keyring password using Gnome's libsecret:
# $ secret-tool store --label=mpop \
#   host pop.freemail.example \
#   service pop3 \
#   user joe.smith

# Password method 2: Store the password in an encrypted file, and tell mpop
# which command to use to decrypt it. This is usually used with GnuPG, as in
# this example. Usually gpg-agent will ask once for the decryption password.
passwordeval gpg2 --no-tty -q -d ~/.mpop-password.gpg

# Password method 3: Store the password directly in this file. Usually it is not
# a good idea to store passwords in cleartext files. If you do it anyway, at
# least make sure that this file can only be read by yourself.
#password secret123

# Password method 4: Store the password in ~/.netrc. This method is probably not
# relevant anymore.

# Password method 5: Do not specify a password. Mpop will then prompt you for
# it. This means you need to be able to type into a terminal when mpop runs.


# A second mail box at the same freemail service
account freemail2 : freemail
user joey


# The POP3 server of your ISP
account isp
host mail.isp.example
auth on
user 12345
# Your ISP runs SpamAssassin, so test each mail for the "X-Spam-Status: Yes"
# header, and delete all mails with this header before downloading them.
filter if [ "`grep "^X-Spam-Status: Yes"`" ]; then exit 1; else exit 0; fi


# Set a default account
account default : freemail
```

9.2 Filtering with SpamAssassin

Use the following to delete all mails that SpamAssassin classifies as spam:

```
filter "/path/to/spamc -c > /dev/null"
```

Since no message body is passed to SpamAssassin, you should disable all body-specific tests in the SpamAssassin configuration file; for example set use_bayes 0.

If your mail provider runs SpamAssassin for you, you just have to check for the result. The following script can do that when used as an mpop filter:

```
#!/bin/sh
if [ "`grep "^X-Spam-Status: Yes"`" ]; then
    exit 1  # kill this message
else
    exit 0  # proceed normally
fi
```

Since the filter command is passed to a shell, all shell constructs are usable, so you can also use this directly:

```
filter if [ "`grep "^X-Spam-Status: Yes"`" ]; then exit 1; else exit 0; fi
```

9.3 Using mpop with Tor

Use the following settings:

```
proxy_host 127.0.0.1
proxy_port 9050
tls on
```

Use an IP address as proxy host name, so that mpop does not leak a DNS query when resolving it.
TLS is required to prevent exit hosts from reading your POP3 session. You also need [tls_trust_file], page 4, or [tls_fingerprint], page 4, to check the server identity.

10 Minimal POP3 server (mpopd)

Mpopd is a minimal POP3 server that delivers mails from a local mailbox in maildir format. It can be used by end users as a way to handle incoming mail via mpop with mail clients that insist on using POP3 (see Section 10.1 [Example using mpopd to handle incoming mail for a POP3-based mail client], page 23).

Mpopd listens on 127.0.0.1 port 1100 by default, but can also run without its own network sockets in inetd mode, where it handles a single POP3 session on standard input / output.

To prevent abuse, mpopd will allow only a limited number of concurrent POP3 sessions, and if an authentication failure occurrs, future authentication requests in any POP3 session will (for a limited duration) only be answered after a small delay.

Mpopd works fine with other programs delivering additional mails into the maildir folders it serves via POP3, but it expects to be the only program to remove or alter mails in these folders. You can e.g. use mpop to deliver new mails into the maildir folder, but you cannot use a mail client to work on the maildir folder at the same time as mpopd.

Mpopd handles the following options:

‘`--version`’
: Print version information

‘`--help`’
: Print help.

‘`--inetd`’
: Start single POP3 session on stdin/stdout

‘`--interface=`*`ip`*’
: Listen on the given IPv6 or IPv4 address instead of 127.0.0.1

‘`--port=`*`number`*’
: Listen on the given port number instead of 1100

‘`--log=`*`none`*`|`*`syslog`*`|`*`filename`*’
: Set logging: none (default), syslog, or logging to the given file.

‘`--auth=`*`user`*`[,`*`passwordeval`*`]`’
: Require authentication with this user name. The password will be retrieved from the given *passwordeval* command (this works just like [passwordeval], page 3, in msmtp) or, if none is given, from the key ring or, if that fails, from a prompt.

‘`--maildir=`*`dir`*’
: Use this maildir as the mailbox.

10.1 Example: using mpopd to handle incoming mail for a POP3-based mail client

Some mail clients cannot get incoming mail from local files and instead insist on using a POP3 server. You can configure mpopd to be that POP3 server and serve your incoming mail from a local maildir folder.

(Similarly, some mail clients cannot send outgoing mail via a program such as msmtp and instead insist on using an SMTP server. You can configure

msmtpd to be that SMTP server and hand the mail over to msmtp. See the corresponding section in the msmtp manual (`https://marlam.de/msmtp/msmtp.html#Example_003a-using-msmtpd-to-handle-outgoing-mail-for-an-SMTP_002dbased-mail-client`).)

For this purpose, mpopd should listen on an unprivileged port, e.g. 1100 (the default). A mailbox is defined using first the '`--auth`' option to set a user name and password and then using the '`--maildir`' option to specify the maildir folder that holds the incoming mail. Multiple such option pairs can be used to define multiple mailboxes, e.g. from different remote mail accounts. Programs such as mpop can deliver new mail into the maildir folders at any time, but as long as mpopd is running no other programs may alter or remove mails from these folders.

Let's use the user name *mpopd-user*. You have two options to manage the password:

1. Store the password in your key ring, e.g. with

   ```
   secret-tool store --label=mpopd host localhost service pop3 user mpopd-user
   ```

   In this case, use the mpopd option '`--auth=mpopd-user`'.
2. Store the password in an encrypted file and use the passwordeval mechanism. Example for gpg:

   ```
   mpopd ... --auth=mpopd-user,'gpg -q -d ~/.mpopd-password.gpg'
   ```

The complete command then is (using the keyring):

```
mpopd --auth=mpopd-user --maildir=/path/to/your/maildir/folder
```

The mail client software must then be configured to use '`localhost`' at port '`1100`' for incoming mail via POP3, and to use authentication with user '`mpopd-user`' and the password you chose. The mail client will probably complain that the POP3 server does not support TLS, but in this special case that is ok since all communication between your mail client and mpopd will stay on the local machine.