

This is the output of mitmdump --help command

I have exported it for convenient reading on web. Feel free to download it and use as you like.

usage: mitmdump [options] [filter]

Args that start with '--' (eg. --conf) can also be set in a config file (~/.mitmproxy/common.conf or ~/.mitmproxy/mitmdump.conf or specified via

--conf) by using .ini or .yaml-style syntax (eg. conf=value). If an arg is specified in more than one place, then command-line values override config

file values which override defaults.

positional arguments:

args

optional arguments:

-h, --help show this help message and exit

--conf CONFIG_FILE config file path

--version show program's version number and exit

--shortversion show program's short version number and exit

--anticache Strip out request headers that might cause the server to return 304-not-modified.

--cadir CADIR Location of the default mitmproxy CA files.
(~/.mitmproxy)

--host Use the Host header to construct URLs for display.

-q, --quiet Quiet.

-r RFILE, --read-flows RFILE

- Read flows from file.
- s "script.py --bar", --script "script.py --bar"
Run a script. Surround with quotes to pass script arguments. Can be passed multiple times.
- t FILTER, --stickycookie FILTER
Set sticky cookie filter. Matched against requests.
- u FILTER, --stickyauth FILTER
Set sticky auth filter. Matched against requests.
- v, --verbose Increase event log verbosity.
- w OUTFILE, --wfile OUTFILE
Write flows to file.
- a OUTFILE, --afile OUTFILE
Append flows to file.
- z, --anticomp Try to convince servers to send us un-compressed data.
- Z SIZE, --body-size-limit SIZE
Byte size limit of HTTP request and response bodies.
Understands k/m/g suffixes, i.e. 3m for 3 megabytes.
- stream SIZE Stream data to the client if response body exceeds the given threshold. If streamed, the body will not be stored in any way. Understands k/m/g suffixes, i.e. 3m for 3 megabytes.
- keep-serving Continue serving after client playback or file read.
We exit by default.
- d, --detail Increase flow detail display level. Can be passed multiple times.

Proxy Options:

- b ADDR, --bind-address ADDR
Address to bind proxy to (defaults to all interfaces)

-I HOST, --ignore HOST

Ignore host and forward all traffic without processing it. In transparent mode, it is recommended to use an IP address (range), not the hostname. In regular mode, only SSL traffic is ignored and the hostname should be used. The supplied value is interpreted as a regular expression and matched on the ip or the hostname. Can be passed multiple times.

--tcp HOST Generic TCP SSL proxy mode for all hosts that match the pattern. Similar to --ignore, but SSL connections are intercepted. The communication contents are printed to the event log in verbose mode.

-n, --no-server Don't start a proxy server.

-p PORT, --port PORT Proxy service port.

-R REVERSE_PROXY, --reverse REVERSE_PROXY

Forward all requests to upstream HTTP server:

`http[s][2http[s]]://host[:port]`

--socks Set SOCKS5 proxy mode.

-T, --transparent Set transparent proxy mode.

-U UPSTREAM_PROXY, --upstream UPSTREAM_PROXY

Forward all requests to upstream proxy server:

`http://host[:port]`

Advanced Proxy Options:

The following options allow a custom adjustment of the proxy behavior. Normally, you don't want to use these options directly and use the provided wrappers instead (-R, -U, -T).

--http-form-in {relative,absolute}

Override the HTTP request form accepted by the proxy

--http-form-out {relative,absolute}

Override the HTTP request form sent upstream by the proxy

Onboarding App:

- `--noapp` Disable the mitmproxy onboarding app.
- `--app-host host` Domain to serve the onboarding app from. For transparent mode, use an IP when a DNS entry for the app domain is not present. Default: mitm.it
- `--app-port 80` Port to serve the onboarding app from.

Client Replay:

- `-c PATH, --client-replay PATH`
Replay client requests from a saved file.

Server Replay:

- `-S PATH, --server-replay PATH`
Replay server responses from a saved file.
- `-k, --kill` Kill extra requests during replay.
- `--rheader RHEADERS` Request headers to be considered during replay.

Can be

passed multiple times.

- `--norefresh` Disable response refresh, which updates times in cookies and headers for replayed responses.
- `--no-pop` Disable response pop from response flow. This makes it possible to replay same response multiple times.
- `--replay-ignore-content`
Ignore request's content while searching for a saved flow to replay
- `--replay-ignore-payload-param REPLAY_IGNORE_PAYLOAD_PARAMS`
Request's payload parameters (application/x-www-form-urlencoded or multipart/form-data) to be ignored while searching for a saved flow to replay. Can be passed multiple times.
- `--replay-ignore-param REPLAY_IGNORE_PARAMS`
Request's parameters to be ignored while searching for

a saved flow to replay. Can be passed multiple times.

--replay-ignore-host Ignore request's destination host while searching for a saved flow to replay

Replacements:

Replacements are of the form "/pattern/regex/replacement", where the separator can be any character. Please see the documentation for more information.

--replace PATTERN Replacement pattern.

--replace-from-file PATH

Replacement pattern, where the replacement clause is a path to a file.

Set Headers:

Header specifications are of the form "/pattern/header/value", where the separator can be any character. Please see the documentation for more information.

--setheader PATTERN Header set pattern.

Proxy Authentication:

Specify which users are allowed to access the proxy and the method used for authenticating them.

--nonanonymous Allow access to any user long as a credentials are specified.

--singleuser USER Allows access to a a single user, specified in the form username:password.

--htpasswd PATH Allow access to users specified in an Apache htpasswd file.

SSL:

- cert SPEC** Add an SSL certificate. SPEC is of the form "[domain=]path". The domain may include a wildcard, and is equal to "*" if not specified. The file at path is a certificate in PEM format. If a private key is included in the PEM, it is used, else the default key in the conf dir is used. The PEM file should contain the full certificate chain, with the leaf certificate as the first entry. Can be passed multiple times.
- cert-forward** Simply forward SSL certificates from upstream.
- ciphers-client CIPHERS_CLIENT**
 Set supported ciphers for client connections. (OpenSSL Syntax)
- ciphers-server CIPHERS_SERVER**
 Set supported ciphers for server connections. (OpenSSL Syntax)
- client-certs CLIENTCERTS**
 Client certificate directory.
- no-upstream-cert** Don't connect to upstream server to look up certificate details.
- ssl-port PORT** Can be passed multiple times. Specify destination ports which are assumed to be SSL. Defaults to [443, 8443].
- ssl-version-client {all,secure,SSLv2,SSLv3,TLSv1,TLSv1_1,TLSv1_2}**
 Set supported SSL/TLS version for client connections. SSLv2, SSLv3 and 'all' are INSECURE. Defaults to secure.
- ssl-version-server {all,secure,SSLv2,SSLv3,TLSv1,TLSv1_1,TLSv1_2}**
 Set supported SSL/TLS version for server connections. SSLv2, SSLv3 and 'all' are INSECURE. Defaults to secure.