mitmproxy

Intercepting HTTP(S) proxy

Front ends

Tool	Use
`mitmproxy`	Interactive terminal UI
`mitmweb`	Web UI on :8081
`mitmdump`	Headless, like tcpdump for HTTP

CONCEPTS

Only intercept traffic from devices and apps you own or are explicitly authorised to inspect.

Three front ends share one engine: mitmproxy (TUI), mitmweb (browser UI) and mitmdump (no UI, scripts and files).

Client proxy set to :8080
↓ HTTP(S)
mitmproxy decrypts with its CA
↓ re-encrypted
Server real destination

Where mitmproxy sits

HTTPS only works once the client trusts the mitmproxy CA; browse to mitm.it through the proxy to install it.

FLOW LIST

j / **k** Move down / up

g / **G**
Go to first / last flow

Enter
Open the focused flow

FLOW LIST (cont.)

q
Back / leave current view

Space Page down

f Set the view filter

m Mark / unmark flow

d Delete flow from view

D Duplicate flow

z Clear the whole flow list

F
Toggle focus follow (tail)

FLOW VIEW

Tab
Switch request / response / detail

e
Edit URL, header, body or method

m
Choose content view (JSON, hex...)

p / **Space**
Previous / next flow

V Revert edits to flow

INTERCEPT AND REPLAY

i
Set intercept filter, e.g. ~u api & ~q

I Toggle intercept on / off

a
Resume the intercepted flow

A
Resume all intercepted flows

r
Replay request to the server

x
Export flow (curl, httpie, raw...)

b
Save response body to file

SAVE, LOAD, GLOBAL

w Save listed flows to file

L Load flows from file

E Open event log

? Context-sensitive help

FILTER EXPRESSIONS

`~u <regex>`
Match URL

`~d <regex>`
Match domain

`~m <regex>`
Match method

`~c <code>`
Match response status code

`~h <regex>`
Match any header

`~b <regex>`
Match any body

`! / & / | / ( )`
Not, and, or, grouping

COMMAND LINE

`mitmproxy -p <port>`
Listen on another port

`mitmweb`
Web interface

`mitmdump -w <file>`
Record flows to file

`mitmproxy -r <file>`
Open a saved flow file

`mitmdump "<filter>"`
Show only matching flows

COMMAND LINE (cont.)

`--mode local[:<app>]`
Capture local apps by name or PID

`--mode reverse:<url>`
Front a single server

`-s <script.py>`
Load an addon script

`--set <option>=<value>`
Set any option

`--ignore-hosts <regex>`
Do not decrypt matching hosts

CERTIFICATES

`http://mitm.it`
Download CA via the proxy

`~/.mitmproxy/mitmproxy-ca-cert.pem`
CA certificate (non-Windows)

`curl --cacert <ca.pem> -x <proxy> <url>`
Trust CA for one curl call

The CA is generated per installation on first start. Remove it from devices when you finish testing.