
mitmproxy-HTTPolice Documentation

Release

Vasiliy Faronov

January 14, 2017

1	Installation	3
2	Usage	5

[mitmproxy](#) is an advanced HTTP debugging tool that can intercept TLS-encrypted connections, supports HTTP/2, and many more.

mitmproxy-HTTPolice is an [inline script](#) for mitmproxy that will check intercepted exchanges and produce an [HTTPolice](#) report. It also works with mitmproxy's command-line tool [mitmdump](#).

For recent changes in mitmproxy-HTTPolice, see the [changelog](#).

Installation

Do this in a Python 3.5+ environment:

```
$ pip3 install mitmproxy-HTTPolice
```

If this is giving you trouble, see [mitmproxy docs](#) and [HTTPolice docs](#) for more detailed instructions.

Note: **Do not use** mitmproxy’s pre-built self-contained binaries. mitmproxy and HTTPolice need to live in the same Python environment, and this is only possible if you install mitmproxy from source via pip. See the “Installation from Source” sections in mitmproxy docs.

Usage

To run HTTPolice together with mitmproxy, use a command like this:

```
$ mitmdump -s "`python3 -m mitmproxy_httppolice` -o html report.html"
```

Note the backticks. Also, you can replace mitmdump with mitmproxy if you wish.

-s is an option for mitmproxy that specifies an inline script to run, along with arguments to that script.

python3 -m mitmproxy_httppolice is a sub-command that prints the path to the script file:

```
$ python3 -m mitmproxy_httppolice  
/home/vasiliy/.local/lib/python3.5/site-packages/mitmproxy_httppolice.py
```

-o html tells HTTPolice to produce HTML reports (omit it if you want a plain text report). Finally, report.html is the name of the output file.

Now, mitmproxy/mitmdump starts up as usual. Every exchange that it intercepts is checked by HTTPolice. When you stop mitmdump (Ctrl+C) or exit mitmproxy, HTTPolice writes an HTML report to report.html.

You can use the -s option to **silence** unwanted notices, just as with the httpolice command-line tool:

```
$ mitmdump -s "`python3 -m mitmproxy_httppolice` -s 1089 -s 1194 report.txt"
```

mitmproxy/mitmdump itself has many interesting options. One of the more useful features is the ability to dump traffic into a file. If you do this, you can then “replay” it as many times as you wish:

```
$ mitmdump --wfile flows.dat  
$ mitmdump --no-server --read-flows flows.dat \  
> -s "`python3 -m mitmproxy_httppolice` /dev/stdout"
```