mitmproxy-HTTPolice Documentation

Release

Vasiliy Faronov

February 28, 2017

Contents

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

mitmproxy-HTTPolice is a script for mitmproxy that will check intercepted exchanges and produce an HTTPolice report. It also works with mitmproxy's companion tools mitmdump and mitmweb.

For recent changes in mitmproxy-HTTPolice, see the changelog.

CHAPTER 1

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.

CHAPTER 2

Usage

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

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

Note the backticks. Replace `mitmdump` with `mitmproxy` or `mitmweb` as needed.

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

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

```
$ python3 -m mitmproxy_httpolice
/home/vasiliy/.local/lib/python3.5/site-packages/mitmproxy_httpolice.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 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_httpolice` -s 1089 -s 1194 report.txt"
```

mitmproxy 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_httpolice` /dev/stdout"
```