mitmproxy-HTTPolice Documentation

Release

Vasiliy Faronov

Aug 02, 2017

Contents

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

Basic usage

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

```
$ mitmdump -s "`python3 -m mitmproxy_httpolice` -o html -w 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, `-w report.html` gives the name of the output file.

Now, mitmdump starts up as usual. Every exchange that it intercepts is checked by HTTPolice. When you stop mitmdump (Ctrl+C), HTTPolice writes a report to `report.html`.

CHAPTER 3

Inspecting traffic on the fly

Often, you don't want to get one big report at the end: you want to see a report for every request/response as it arrives. You can do this with the `--tail` option, which tells mitmproxy-HTTPolice to regenerate the report on every new exchange:

```
$ mitmdump -s "`python3 -m mitmproxy_httpolice` -o html -w report.html --tail 5"
```

With the above command, `report.html` will always contain a report on the **last 5 exchanges** seen by mitmproxy. The latest exchange is at the **bottom** of the page.

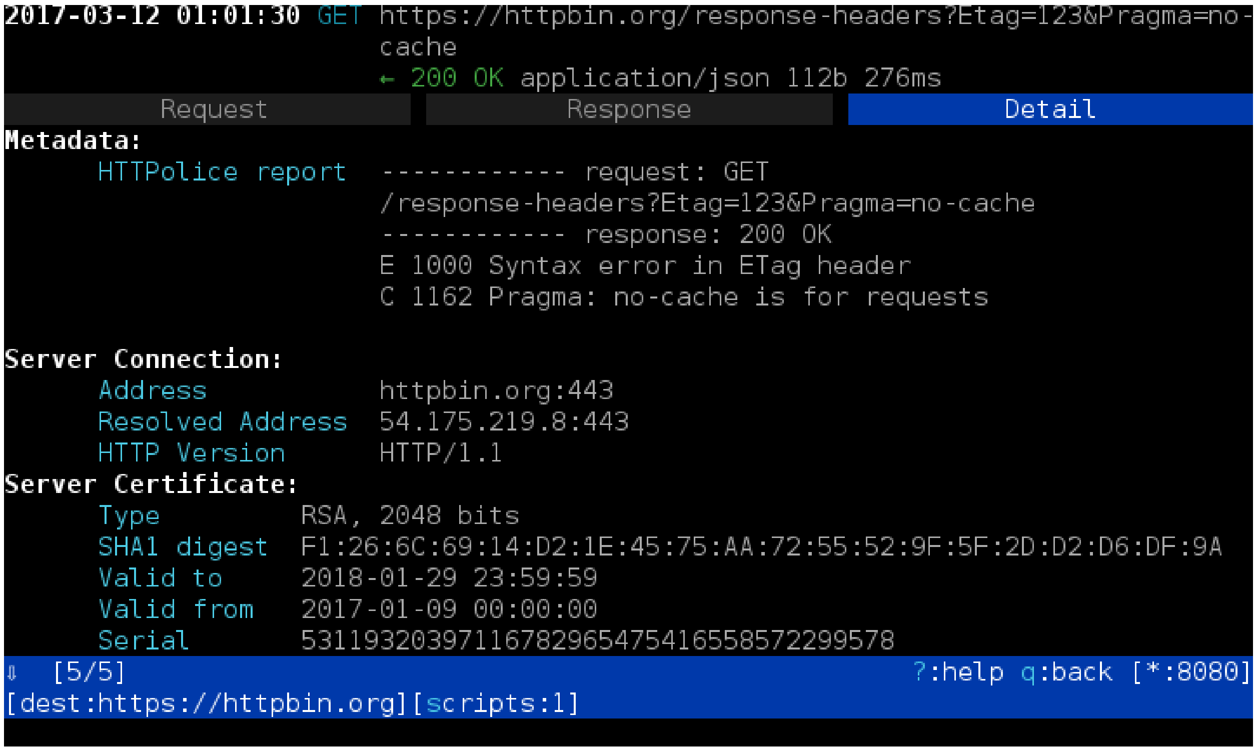
Instead of constantly refreshing that page, you can keep an eye on the log that mitmdump prints to the console, because HTTPolice will notify you whenever there's something to see:

```
HTTPolice found 2 errors, 3 comments in: POST /api/v1/customer/ - 201 Created
```

CHAPTER 4

Integration with the console UI

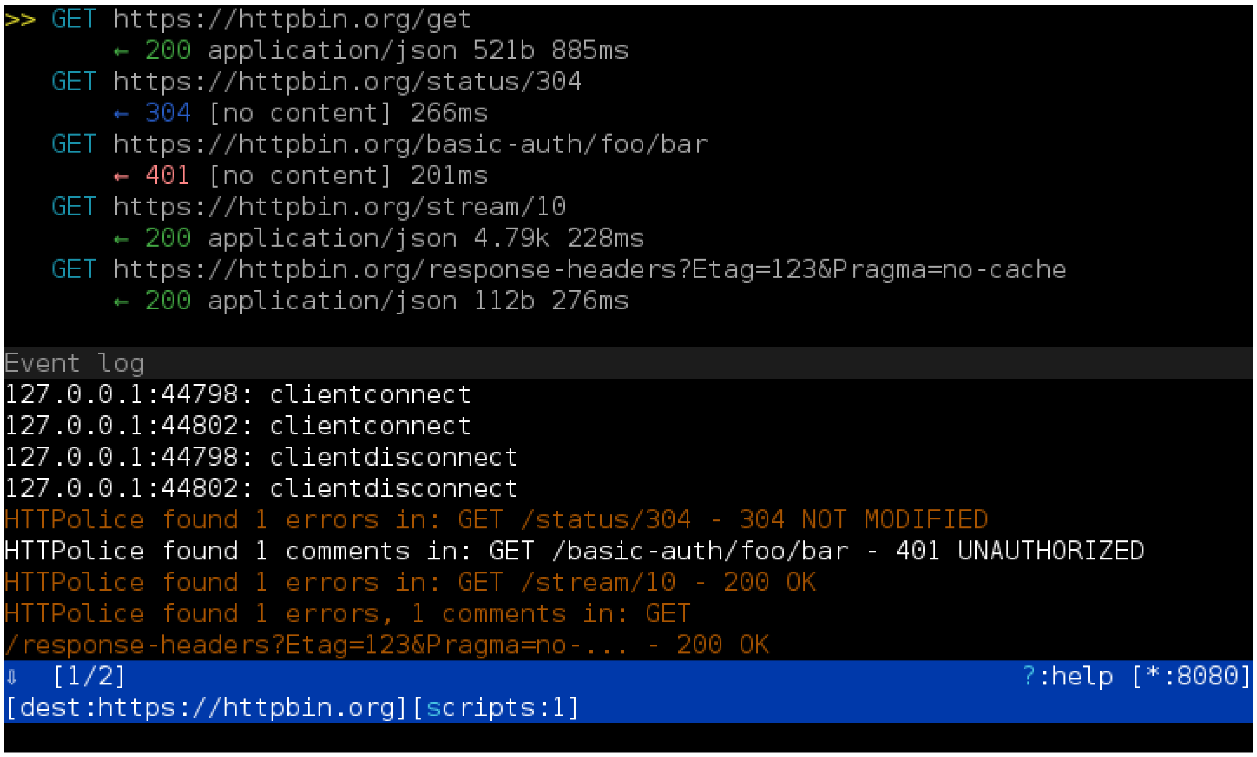
When using the console UI of mitmproxy (the tool named `mitmproxy`), you can also see the report for every exchange (“flow” in mitmproxy parlance) on its “Detail” tab:



```
2017-03-12 01:01:30 GET https://httpbin.org/response-headers?Etag=123&Pragma=no-
                        cache
                        ← 200 OK application/json 112b 276ms
          Request                   Response                    Detail
Metadata:
      HTTPolice report   ------------ request: GET
                         /response-headers?Etag=123&Pragma=no-cache
                         ------------ response: 200 OK
                         E 1000 Syntax error in ETag header
                         C 1162 Pragma: no-cache is for requests

Server Connection:
      Address            httpbin.org:443
      Resolved Address   54.175.219.8:443
      HTTP Version       HTTP/1.1
Server Certificate:
      Type          RSA, 2048 bits
      SHA1 digest   F1:26:6C:69:14:D2:1E:45:75:AA:72:55:52:9F:5F:2D:D2:D6:DF:9A
      Valid to      2018-01-29 23:59:59
      Valid from    2017-01-09 00:00:00
      Serial        53119320397116782965475416558572299578
⇩  [5/5]                                              ?:help q:back [*:8080]
[dest:https://httpbin.org][scripts:1]
```

How do you even know that there’s anything to see there? Currently the only way is to follow the event log, which you can trigger by pressing the ‘e’ key:



```
>> GET https://httpbin.org/get
       ← 200 application/json 521b 885ms
   GET https://httpbin.org/status/304
       ← 304 [no content] 266ms
   GET https://httpbin.org/basic-auth/foo/bar
       ← 401 [no content] 201ms
   GET https://httpbin.org/stream/10
       ← 200 application/json 4.79k 228ms
   GET https://httpbin.org/response-headers?Etag=123&Pragma=no-cache
       ← 200 application/json 112b 276ms

Event log
127.0.0.1:44798: clientconnect
127.0.0.1:44802: clientconnect
127.0.0.1:44798: clientdisconnect
127.0.0.1:44802: clientdisconnect
HTTPolice found 1 errors in: GET /status/304 - 304 NOT MODIFIED
HTTPolice found 1 comments in: GET /basic-auth/foo/bar - 401 UNAUTHORIZED
HTTPolice found 1 errors in: GET /stream/10 - 200 OK
HTTPolice found 1 errors, 1 comments in: GET
/response-headers?Etag=123&Pragma=no-... - 200 OK
⇩  [1/2]                                                       ?:help [*:8080]
[dest:https://httpbin.org][scripts:1]
```

When using mitmproxy-HTTPolice like this, you don't have to specify an output file. You can simply run:

```
$ mitmproxy -s "`python3 -m mitmproxy_httpolice`"
```

Of course, if you *also* want a fully-fledged report, you can combine this with the `-w`, `-o` and `--tail` options.

CHAPTER 5

More options

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 -w 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` -w /dev/stdout"
```