教你用 Python 操控你的上网请求

原创 極光 Python技术 3月4日

今天给大家介绍个有意思的工具，不知道你有没有听说过中间人攻击（Man-in-the-middle attack）简称 MITM，是一种“间接”的入侵攻击，这种攻击模式是通过各种技术手段将受入侵者控制的一台计算机虚拟放置在网络连接中的两台通信计算机之间，这台计算机就称为“中间人”，看下这张图可能更容易理解。

mitmproxy

好了，下面就开始我们介绍我们今天的主角 `mitmproxy` ，它和其他抓包工具相比，不仅可以截获请求帮助开发者查看、分析，更可以通过 Python 自定义脚本进行二次开发。而且由于 `mitmproxy` 工作在 HTTP 层，而且现在客户端拥有了检测并规避中间人攻击的能力，所以并不会真的对无辜的人发起中间人攻击，只能用来做开发或测试。好了，接下来我们就开始一起看他到底有什么本事。

安装

安装还是很简单的，只需要用 `pip` 直接自动安装就可以了，执行以下安装命令：

```
$ pip3 install mitmproxy
```

如果没有提示出错，就算是安装成功了。然后我们在分别运行以下三个命令，可以展示出相应的版本信息。

```
$ mitmproxy --version
Mitmproxy: 5.0.1
Python:    3.7.4
OpenSSL:   OpenSSL 1.1.0j  20 Nov 2018
Platform:  Darwin-16.7.0-x86_64-i386-64bit

$ mitmdump --version
Mitmproxy: 5.0.1
Python:    3.7.4
OpenSSL:   OpenSSL 1.1.0j  20 Nov 2018
Platform:  Darwin-16.7.0-x86_64-i386-64bit

$ mitmweb --version
Mitmproxy: 5.0.1
Python:    3.7.4
OpenSSL:   OpenSSL 1.1.0j  20 Nov 2018
Platform:  Darwin-16.7.0-x86_64-i386-64bit
```

mitmproxy、mitmdump、mitmweb 区别

其实 mitmproxy、mitmdump、mitmweb 这三个功能本质是一样的，区别主要是它们的交互的方式不同。

- mitmproxy：主要是以控制台的方式交互
- mitmdump：主要是以命令行的方式交互
- mitmweb：主要以 web 的形式进行交互

在这里可以看出，以 web 方式交互对我们来说应该是最简单的，接下来就以 mitmweb 为基础介绍它的功能。

启动 mitmweb

直接执行命令 mitmweb 就可以启动，启动以后的效果如下，服务会监听本机 8080 端口，并且通过 8081端口可以访问 web 控制台：

```
$ mitmweb
Web server listening at http://127.0.0.1:8081/
Proxy server listening at http://*:8080
```

我们在浏览器地址输入：127.0.0.1:8081，回车后返回如下页面：

配置浏览器代理

服务启动完成，现在就需要配置浏览器通过本地服务 8080 端口来联网，从而使 mitm 达到做为“中间人”的目的。由于只是测试一下，所以可以使用命令行启动浏览器，这里我用的 Google Chrome，其他浏览器也差别不大，输入以下命令即可：

Linux:

> google-chrome --proxy-server=127.0.0.1:8080 --ignore-certificate-errors

Windows:

> D:/google-chrome.exe --proxy-server=127.0.0.1:8080 --ignore-certificate-errors

MacOS:

> open -a /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --args -proxy-server=127.0.0.1:8080 -ignore-certificate-errors

浏览器启动完成后，会有一个安全提示，如下图所示：

mitmproxy 的使用

下面我们以访问百度首页为例看下 mitmproxy 在中间是如何修改数据的。首先我们在 mitmweb 页面配置下只拦截 baidu 相关的请求，如下图所示：

现在我们通过浏览器再请求一下百度，这时我们再看下 mitmweb 页面会出现黄色的请求提示，表明我们的请求已经被 mitmproxy 拦截，然后我们就可以通过点击图中靠右边的小铅笔图标，就可以修改我们拦截的请求信息了，请看下图：

当修改完需要把拦截的请求放行，则需要单击工具栏中 `Resume` (绿色图标)按钮即可。请求信息发出去后，我们很快就会收到服务器返回的信息，返回的信息也一样会被 `mitmproxy` 拦截，并且我们也可以对返回的信息进行修改，返回信息如下图，多了个 `Response` 标签页面：

好了，在这里我试着改了下返回页面的 `title`，如下图：

然后我们再单击 `Resume` 按钮，将返回信息放行，就会在浏览器看到如下图的效果：

好了，以上我们只是手动简单操作了下，主要是可以让大家明白它的原理，接下来我们就来演示，通过 Python 脚本来自动完成数据的操作。

Python 脚本示例

接下来我们通过 Python 写一段脚本，实现在你用百度搜索任意内容时，都把你搜索的内容改为“建议使用Google搜索”，并且把请求返回内容里面，所有“百度”的字串都自动替换为“谷歌”，这个小脚本让我们同时实现了修改请求和返回内容，现在就上代码：

```python
# baidu.py

# 引入对应模块
import mitmproxy.http
from mitmproxy import ctx, http

class Baidu:
    # 请求时需要处理
    def request(self, flow: mitmproxy.http.HTTPFlow):
        if flow.request.host != "www.baidu.com" or not flow.request.path.startswith("/s"):
            return

```

```
        if "wd" not in flow.request.query.keys():
            ctx.log.warn("can not get search word from %s" % flow.request.pretty_url)
            return

        # 打印日志
        ctx.log.info("正在搜索: %s" % flow.request.query.get("wd"))

        # 替换搜索关键词
        flow.request.query.set_all("wd", ["建议使用Google搜索"])

    # 请求返回时需要处理
    def response(self, flow: mitmproxy.http.HTTPFlow):
        # 获取请求返回的文本并替换
        text = flow.response.get_text()
        text = text.replace("百度", "谷歌")
        flow.response.set_text(text)

# 增加插件
addons = [
    Baidu()
]
```

写好代码保存，接下来我们重新启动 `mitmweb`，这次命令后需要加上参数使 Python 脚本生效，在命令行输入：`mitmweb -s baidu.py`，启动完成后我们回到浏览器打开百度随便输入个内容进行搜索，你会看到效果如下图所示：

搜索前

搜索后

可以看到上面我们随便个词点搜索，返回的结果都是“建议使用Google搜索”，并且右上角“百度”也都被替换为了“谷歌”。

总结

本文为大家介绍了 `mitmproxy` 工具的安装以及如何使用，并写了一段小脚本简单实现了用它如何修改拦截的请求和返回的内容。当然它的功能不止这么简单，有了它我们可以做很多事情，有兴趣的话后续再为大家介绍。

> 文中示例代码：https://github.com/JustDoPython/python-100-day

参考

https://mitmproxy.org

PS： 公号内回复「Python」即可进入 Python 新手学习交流群，一起 **100天计划！**

-END-

Python 技术

关于 **Python** 都在这里