

Forum:[Hack Forums](#)

Thread: [\[sunjester tuts\] Proxy Rotation with Python](#)

realsunjester - March 10, 2019, 8:43 am

Introduction

All of my tutorials are written in a unix environment. If you are using Kali, perfect. I am using Debian (xenial). If you have any errors, post them here and I'll see what I can do. You will need Python and probably will need the requests library installed.

Requests

You can install the requests library with pip

Code pip install requests

and you can import the library using the import keyword.

Code `#!/usr/bin/env python`

`import requests`
`Proxies`

Using a [proxy with the requests library](#) is easy.

Code `#!/usr/bin/env python`

`import requests`

```
proxies = {  
    "socks5": "67.205.174.209:1080"  
}
```

```
s = requests.get("https://api.ipify.org", proxies=proxies)  
print s.content
```

Rotation

The theory here is that you "rotate" through proxies as you make requests. You could randomly select a proxy from the array or even do it by timeout. The most simple way is to just iterate through the array as you make the requests, by default, I believe that is what the the requests lib does already. So let's pick a random one.

Code `#!/usr/bin/env python`

`import requests, random, os`

```
proxies = [  
    "163.172.220.221:8888",  
    "138.68.41.90:3128"  
]
```

```
proxy = random.choice(proxies)  
os.environ['HTTP_PROXY'] = proxy
```

```
s = requests.get("https://api.ipify.org")  
print s.content
```

The script above uses the random library to pick a random one from the array and then uses the os library to set an environment variable to the proxy we randomly selected.

AeroBlue - March 10, 2019, 2:31 pm

The problem that I always have is usually big proxy lists come in a mix of http and https and you have to explicitly say which one it is, or else it will throw an error. Any way around this issue?

Vassilios - March 10, 2019, 7:30 pm

(Mar 10, 2019 02:31 PM)Aero Blue Wrote: [\[CITE\]](#) The problem that I always have is usually big proxy lists come in a mix of http and https and you have to explicitly say which one it is, or else it will throw an error. Any way around this issue?

You could make it try one and if it fails try the other.

realsunjester - March 11, 2019, 12:33 pm

(Mar 10, 2019 02:31 PM)Aero Blue Wrote: [\[CITE\]](#) The problem that I always have is usually big proxy lists come in a mix of http and https and you have to explicitly say which one it is, or else it will throw an error. Any way around this issue?

you could test if its http or https with a curl request, determining where to sort it.

so this will work if its http

Code curl -x http://123.456.789.012:8080 https://api.ipify.org/

and if its https this will work

Code curl -x https://123.456.789.012:8080 https://api.ipify.org/

Athr - March 12, 2019, 11:24 pm

I'm using something like this:

Code def get_response(x):

try:

req = requests.post('https://example.com/' + x, headers=headers, params=params, cookies=cookies, data=data, proxies = {'http': proxies[0]})

except Exception as e:

print(e)

print("bad proxy. using another one...")

proxies.pop(0)

return get_response(x)

return req.text

Let me know if there is a better solution.

darksh33p - March 18, 2019, 8:56 pm

would be nice to add a failure counter to the proxies dictionary so you can know when to quit a proxy. depend on what yer doing

realsunjester - March 18, 2019, 9:49 pm

(Mar 18, 2019 08:56 PM)darksh33p Wrote: [\[CITE\]](#) would be nice to add a failure counter to the proxies dictionary so you can know when to quit a proxy. depend on what yer doing

yea i thought i put something in there about a timeout but i guess not. im working on a book right now, i'll add this.