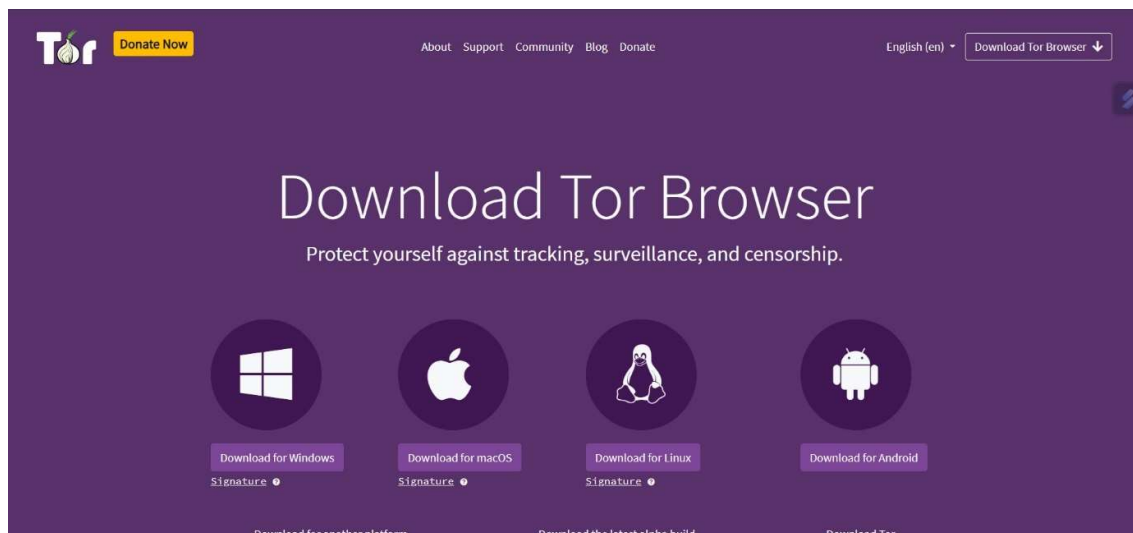


Anonymizing using proxies in TOR

Introduction:

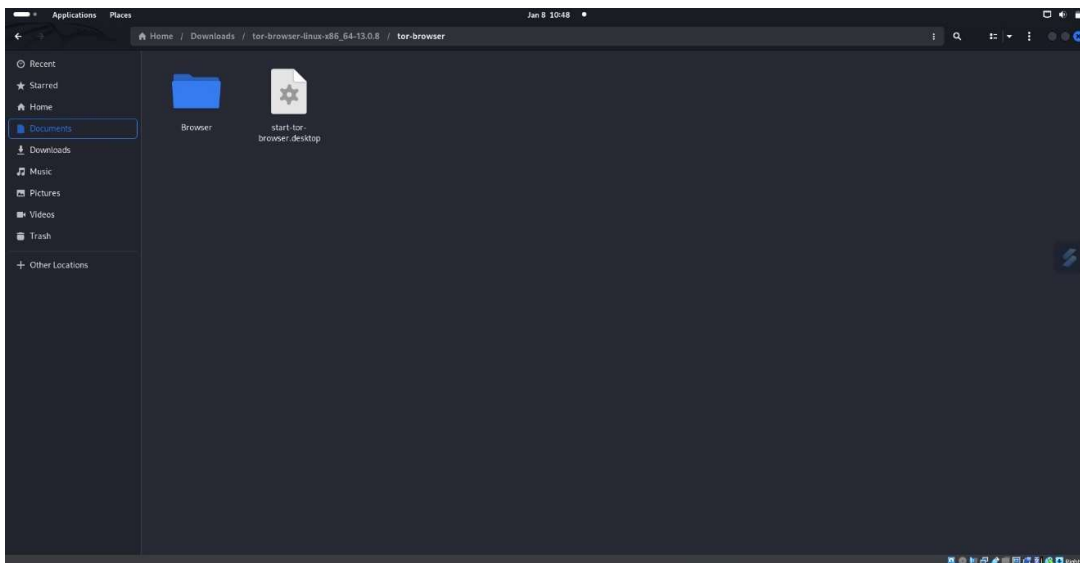
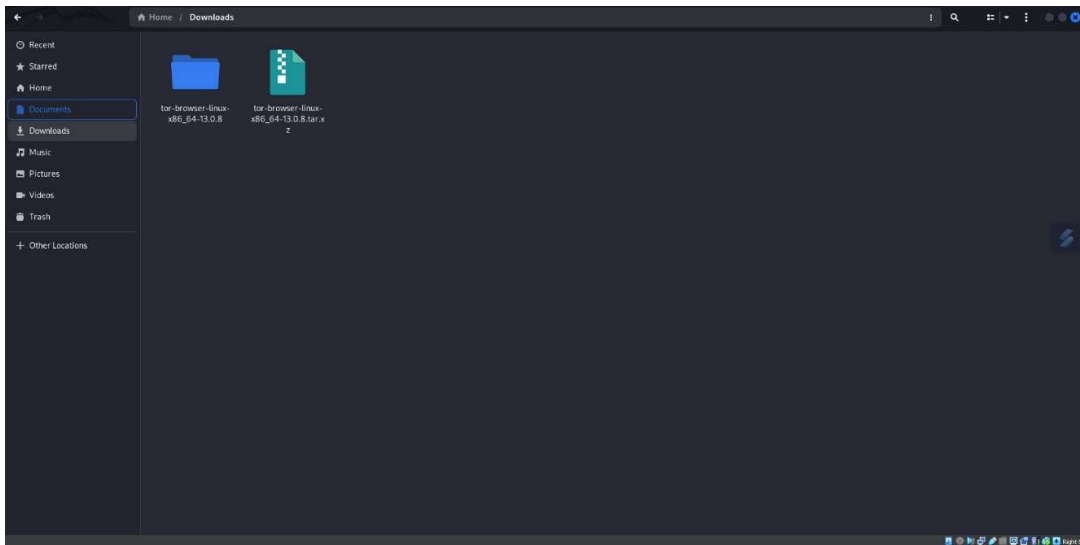
Tor, short for The Onion Router, is free and open-source software for enabling anonymous communication. It directs Internet traffic via a free, worldwide, volunteer overlay network that consists of more than seven thousand relays. Using Tor makes it more difficult to trace a user's Internet activity.

INSTALLATION

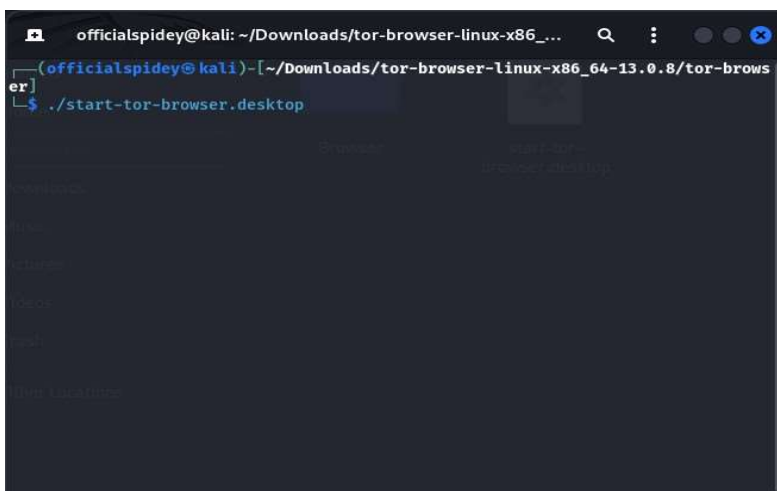


Firstly, for installation of tor, we need to visit the official <https://www.torproject.org> and then proceed to the installation step.

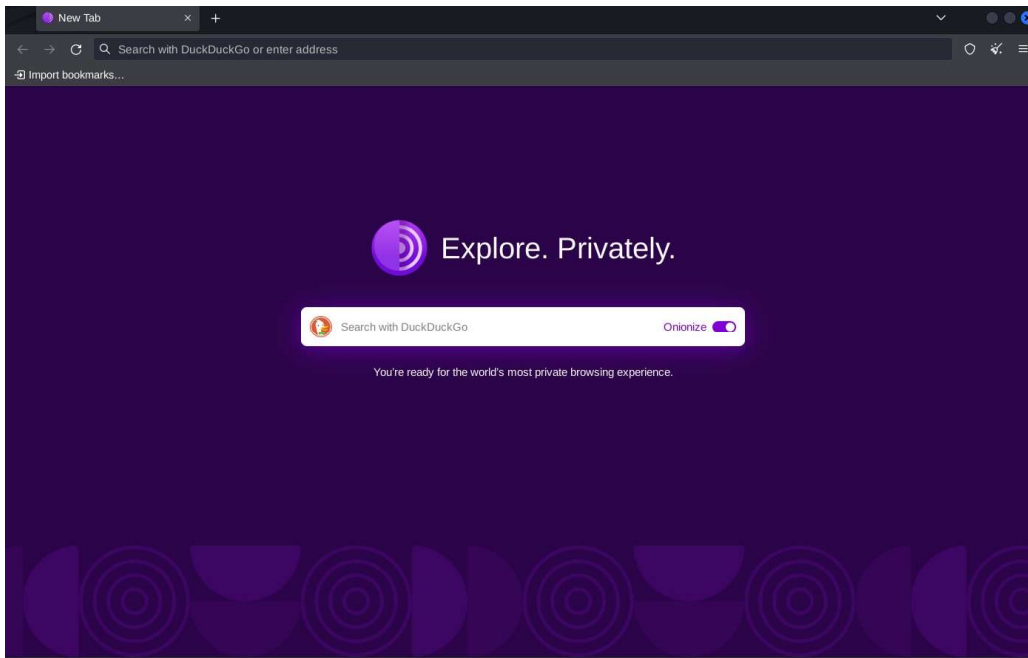
Then we have a folder with the folder and the “.xz” file. Then we have to extract the file so that we get all the elements for the tor.



Here, in this step, open the terminal and run the following



After this step, we will be headed to the tor browser site



After this step, we need to anonymize more. This step could be achieved through adding the proxy servers. Firstly, we need to get some socks4 or socks5 proxies.

SOCKS4 and SOCKS5 are two versions of the SOCKS protocol, an internet protocol that allows one computer to connect to another via a third computer. SOCKS4 is an older protocol that provides basic proxy functionality but lacks support for advanced features like authentication and UDP. SOCKS5 is an upgraded version that offers improved security and authentication. SOCKS5 also supports various types of traffic, including TCP, UDP, and IPv6.

So, for better security, we opt the socks5 proxies.

The proxies we opted for are:

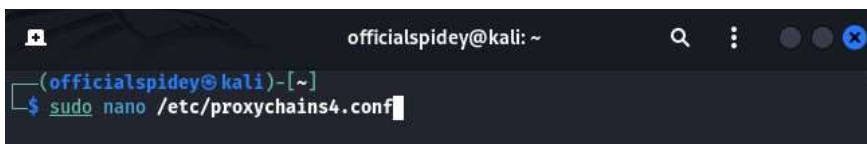
socks5 192.111.134.10 4145

socks5 192.111.137.37 18762

socks5 162.240.72.139 59011

socks5 51.161.99.114 29758

These proxies should be inputted into the proxychains configuration file.



Then input the proxies in the proxy list place:

```
officialspidey@kali: ~  
GNU nano 2.2 /etc/proxychains.conf  
# Trying to proxy connections to destinations which are dnatted,  
# will result in proxying connections to the new given destinations.  
# Whenever I connect to 1.1.1.1 on port 1234 actually connect to 1.1.1.2 on port 443  
# dnatt 1.1.1.1:1234 1.1.1.2:443  
  
# Whenever I connect to 1.1.1.1 on port 443 actually connect to 1.1.1.2 on port 443  
# (no need to write :443 again)  
# dnatt 1.1.1.2:443 1.1.1.2  
  
# No matter what port I connect to on 1.1.1.1 port actually connect to 1.1.1.2 on port 443  
# dnatt 1.1.1.1 1.1.1.2:443  
  
# Always, instead of connecting to 1.1.1.1, connect to 1.1.1.2  
# dnatt 1.1.1.1 1.1.1.2  
  
# Proxylist format  
# type ip port [user pass]  
# (values separated by 'tab' or 'blank')  
# only numeric ipv4 addresses are valid  
  
# Examples:  
  
# socks5 192.168.07.78 1080 lamer secret  
# http 192.168.09.3 8080 justa hidden  
# socks4 192.168.1.49 1080  
# http 192.168.39.93 8080  
  
# proxy types: http, socks4, socks5, raw  
# * raw: The traffic is simply forwarded to the proxy without modification.  
# (auth types supported: "basic"-http "user/pass"-socks )  
  
[Proxylist]  
# add proxy here ...  
# example  
# defaults set to "raw"  
socks4 127.0.0.1 9050  
socks5 127.0.0.1 9050
```

```
officialspidey@kali: ~  
GNU nano 2.2 /etc/proxychains.conf  
# Trying to proxy connections to destinations which are dnatted,  
# will result in proxying connections to the new given destinations.  
# Whenever I connect to 1.1.1.1 on port 1234 actually connect to 1.1.1.2 on port 443  
# dnatt 1.1.1.1:1234 1.1.1.2:443  
  
# Whenever I connect to 1.1.1.1 on port 443 actually connect to 1.1.1.2 on port 443  
# (no need to write :443 again)  
# dnatt 1.1.1.2:443 1.1.1.2  
  
# No matter what port I connect to on 1.1.1.1 port actually connect to 1.1.1.2 on port 443  
# dnatt 1.1.1.1 1.1.1.2:443  
  
# Always, instead of connecting to 1.1.1.1, connect to 1.1.1.2  
# dnatt 1.1.1.1 1.1.1.2  
  
# Proxylist format  
# type ip port [user pass]  
# (values separated by 'tab' or 'blank')  
# only numeric ipv4 addresses are valid  
  
# Examples:  
  
# socks5 192.168.07.78 1080 lamer secret  
# http 192.168.09.3 8080 justa hidden  
# socks4 192.168.1.49 1080  
# http 192.168.39.93 8080  
  
# proxy types: http, socks4, socks5, raw  
# * raw: The traffic is simply forwarded to the proxy without modification.  
# (auth types supported: "basic"-http "user/pass"-socks )  
  
[Proxylist]  
socks5 192.111.134.10 4345  
socks5 192.111.137.17 18762  
socks5 162.240.72.139 59011  
socks5 51.161.99.114 29758  
# add proxy here ...  
# example  
# defaults set to "raw"  
socks4 127.0.0.1 9050  
socks5 127.0.0.1 9050
```

And then, save it by pressing “ctrl + o” and then press “enter” following by “ctrl +x” to exit.

And then for more anonymity, we can also opt the method of using the VPN.

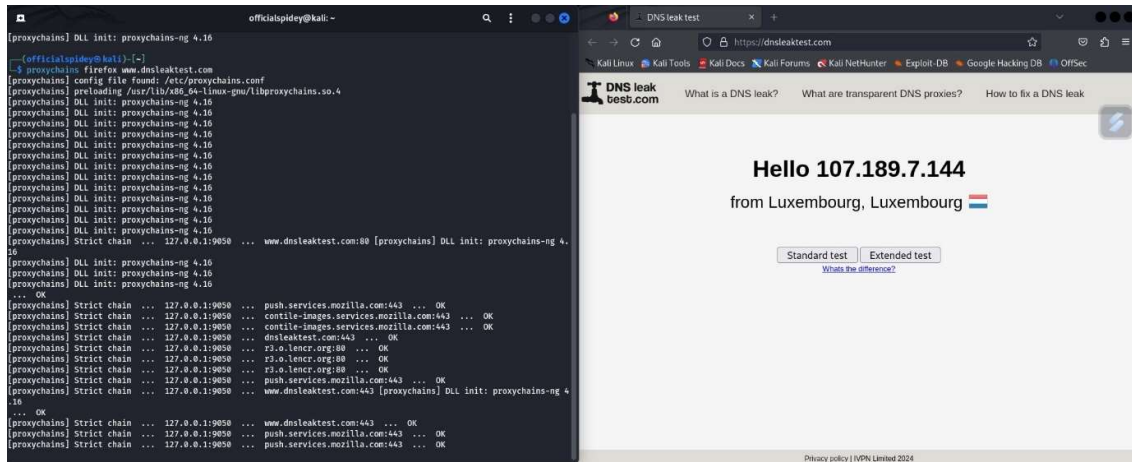
For this we need to install a VPN and then configure it.

After all this process, for checking the IP of our system (so to verify our anonymity level), we need to access the website www.dnsleaktest.com , which gives us a reliable idea about our anonymity levels.

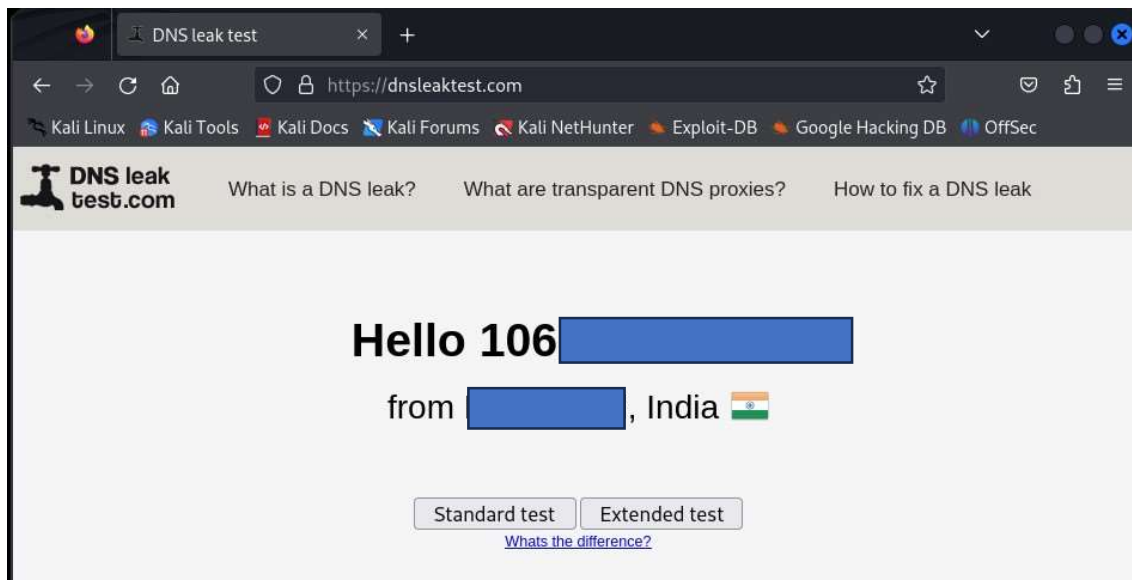
The above steps can be done as:

```
(officialspidey@kali)-[~]  
$ proxychains firefox www.dnsleaktest.com
```

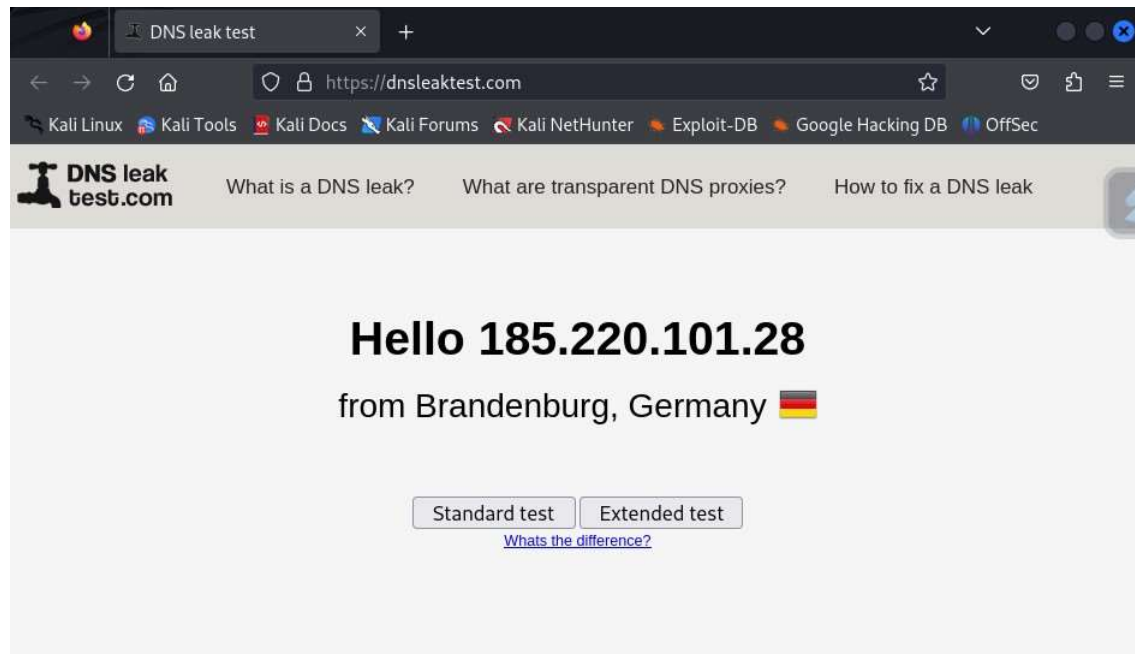
After doing this, we get something like this



The actual IP of our system by default is given as



After following the same exact process as done above, once again after testing in the “dnsleaktest” we get



In this way, the anonymity is maintained for our system such that no hacker or any illegal intruder can trace our systems.