Network & System Security

Exercise 4

Prerequisites:

- Git
- C compiler
- GNU make
- libevent-dev
- libssl-dev

sudo apt-get install git build-essential automake libevent-dev libssl-dev zlib1g-dev

References -

https://tor.stackexchange.com/questions/75/how-can-i-install-tor-from-the-source-code-in-the-git-repository

https://www.tecmint.com/use-tor-network-in-web-browser/

https://pypi.org/project/stem/

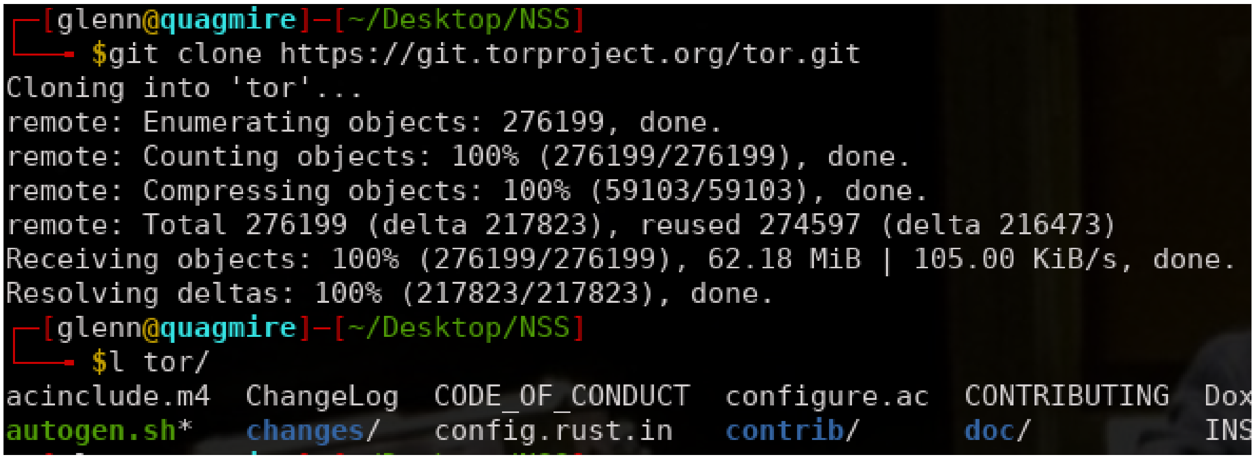
https://metrics.torproject.org/rs.html#details/E34C28D652520D7C8D386EA3958EA924910E647B

https://metrics.torproject.org/rs.html#search/country:in

Part 1 : Running Tor Client

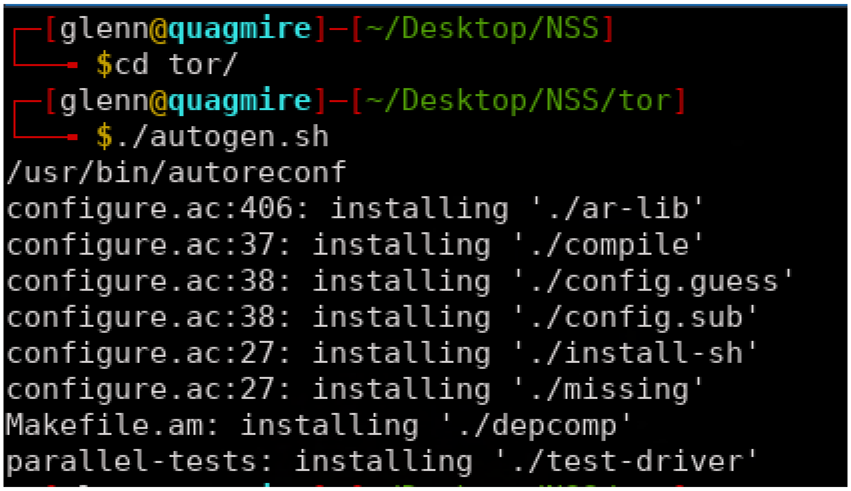
STEP 1 : Installing tor from source and configuring firefox

- Clone Tor from git.torproject.org/tor.git : git clone https://git.torproject.org/tor.git



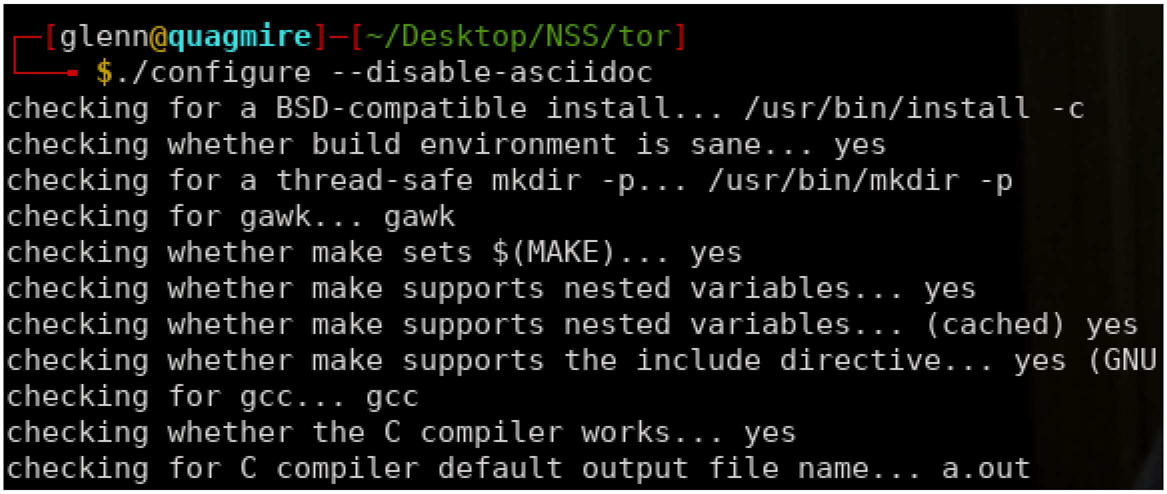
```
┌─[glenn@quagmire]─[~/Desktop/NSS]
└──╼ $git clone https://git.torproject.org/tor.git
Cloning into 'tor'...
remote: Enumerating objects: 276199, done.
remote: Counting objects: 100% (276199/276199), done.
remote: Compressing objects: 100% (59103/59103), done.
remote: Total 276199 (delta 217823), reused 274597 (delta 216473)
Receiving objects: 100% (276199/276199), 62.18 MiB | 105.00 KiB/s, done.
Resolving deltas: 100% (217823/217823), done.
┌─[glenn@quagmire]─[~/Desktop/NSS]
└──╼ $l tor/
acinclude.m4  ChangeLog  CODE_OF_CONDUCT  configure.ac  CONTRIBUTING  Dox
autogen.sh*   changes/   config.rust.in   contrib/      doc/          INS
```

- Change to tor directory and run autogen bash script

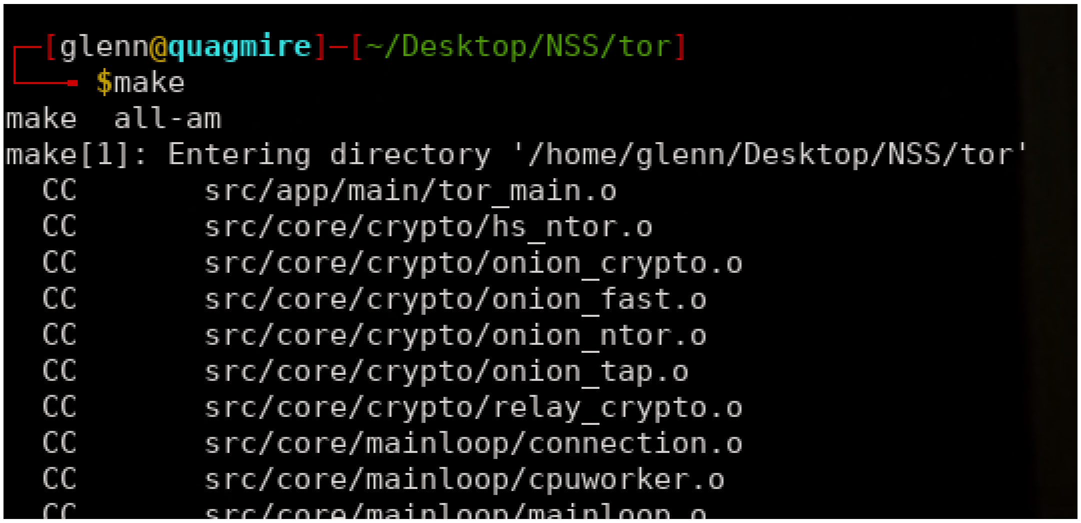


```
┌─[glenn@quagmire]─[~/Desktop/NSS]
└──╼ $cd tor/
┌─[glenn@quagmire]─[~/Desktop/NSS/tor]
└──╼ $./autogen.sh
/usr/bin/autoreconf
configure.ac:406: installing './ar-lib'
configure.ac:37: installing './compile'
configure.ac:38: installing './config.guess'
configure.ac:38: installing './config.sub'
configure.ac:27: installing './install-sh'
configure.ac:27: installing './missing'
Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
```

- Configure tor using ./configure and then compile it using make command

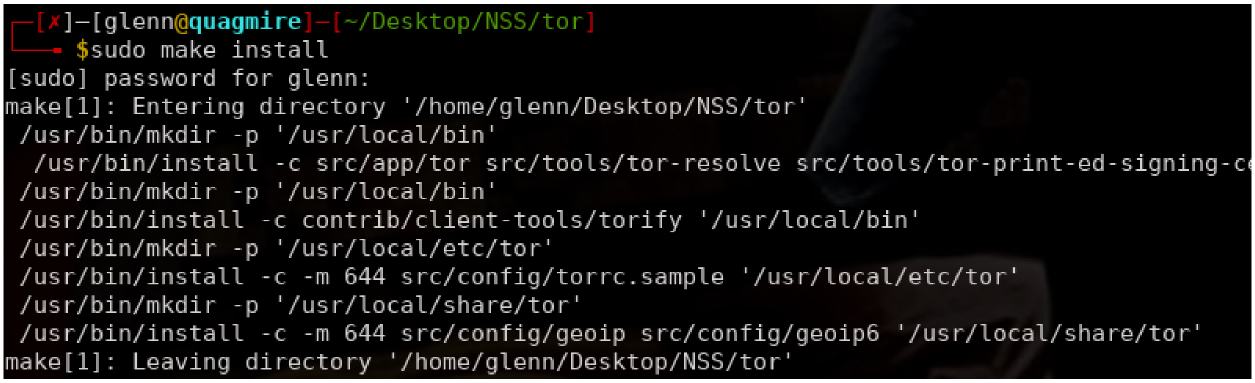


```
┌─[glenn@quagmire]─[~/Desktop/NSS/tor]
└──╼ $./configure --disable-asciidoc
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports nested variables... (cached) yes
checking whether make supports the include directive... yes (GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
```



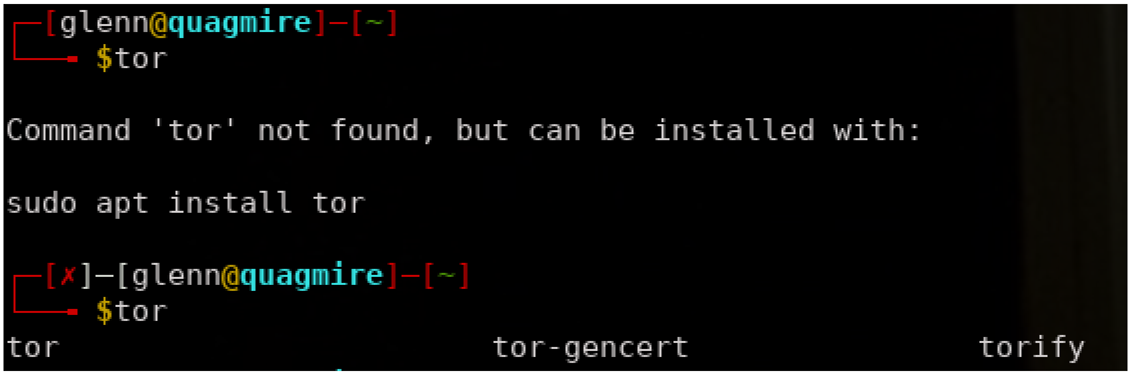
```
┌─[glenn@quagmire]─[~/Desktop/NSS/tor]
└──╼ $make
make  all-am
make[1]: Entering directory '/home/glenn/Desktop/NSS/tor'
  CC       src/app/main/tor_main.o
  CC       src/core/crypto/hs_ntor.o
  CC       src/core/crypto/onion_crypto.o
  CC       src/core/crypto/onion_fast.o
  CC       src/core/crypto/onion_ntor.o
  CC       src/core/crypto/onion_tap.o
  CC       src/core/crypto/relay_crypto.o
  CC       src/core/mainloop/connection.o
  CC       src/core/mainloop/cpuworker.o
  CC       src/core/mainloop/mainloop.o
```

- Finally, use make install



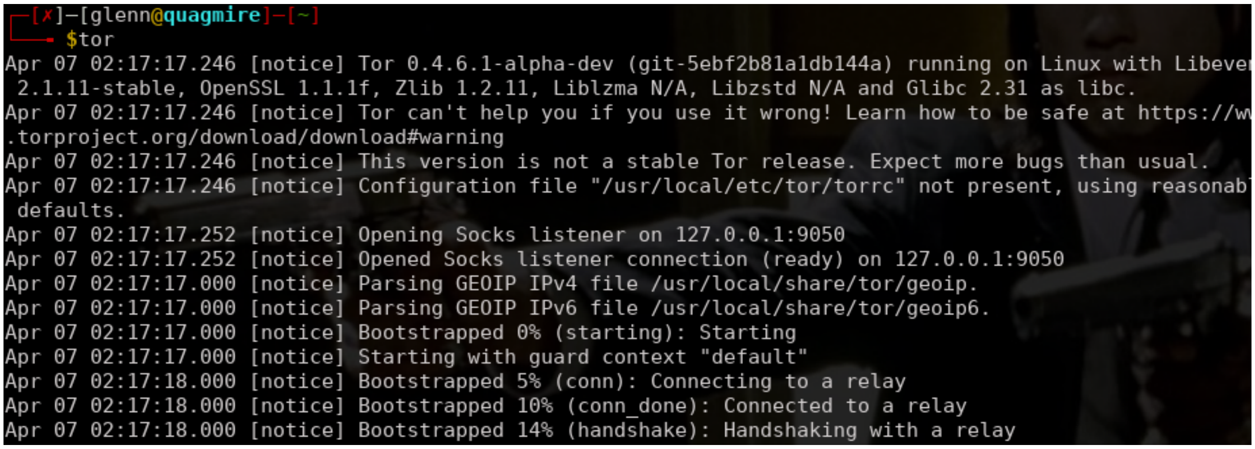
```
┌─[✗]─[glenn@quagmire]─[~/Desktop/NSS/tor]
└──╼ $sudo make install
[sudo] password for glenn:
make[1]: Entering directory '/home/glenn/Desktop/NSS/tor'
 /usr/bin/mkdir -p '/usr/local/bin'
  /usr/bin/install -c src/app/tor src/tools/tor-resolve src/tools/tor-print-ed-signing-ce
 /usr/bin/mkdir -p '/usr/local/bin'
 /usr/bin/install -c contrib/client-tools/torify '/usr/local/bin'
 /usr/bin/mkdir -p '/usr/local/etc/tor'
 /usr/bin/install -c -m 644 src/config/torrc.sample '/usr/local/etc/tor'
 /usr/bin/mkdir -p '/usr/local/share/tor'
 /usr/bin/install -c -m 644 src/config/geoip src/config/geoip6 '/usr/local/share/tor'
make[1]: Leaving directory '/home/glenn/Desktop/NSS/tor'
```

The screenshot below is when “tor” command was issued before and after the installation, it shows that tor has been built successfully from scratch



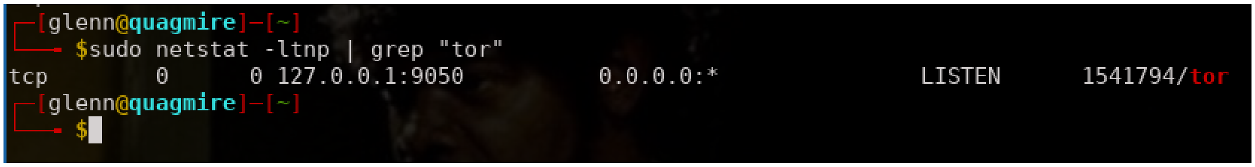
```
┌─[glenn@quagmire]─[~]
└──╼ $tor

Command 'tor' not found, but can be installed with:

sudo apt install tor

┌─[✗]─[glenn@quagmire]─[~]
└──╼ $tor
tor                           tor-gencert                   torify
```

- Now configuring the firefox browser manually by changing proxy settings to use tor program:
    - First run the tor program



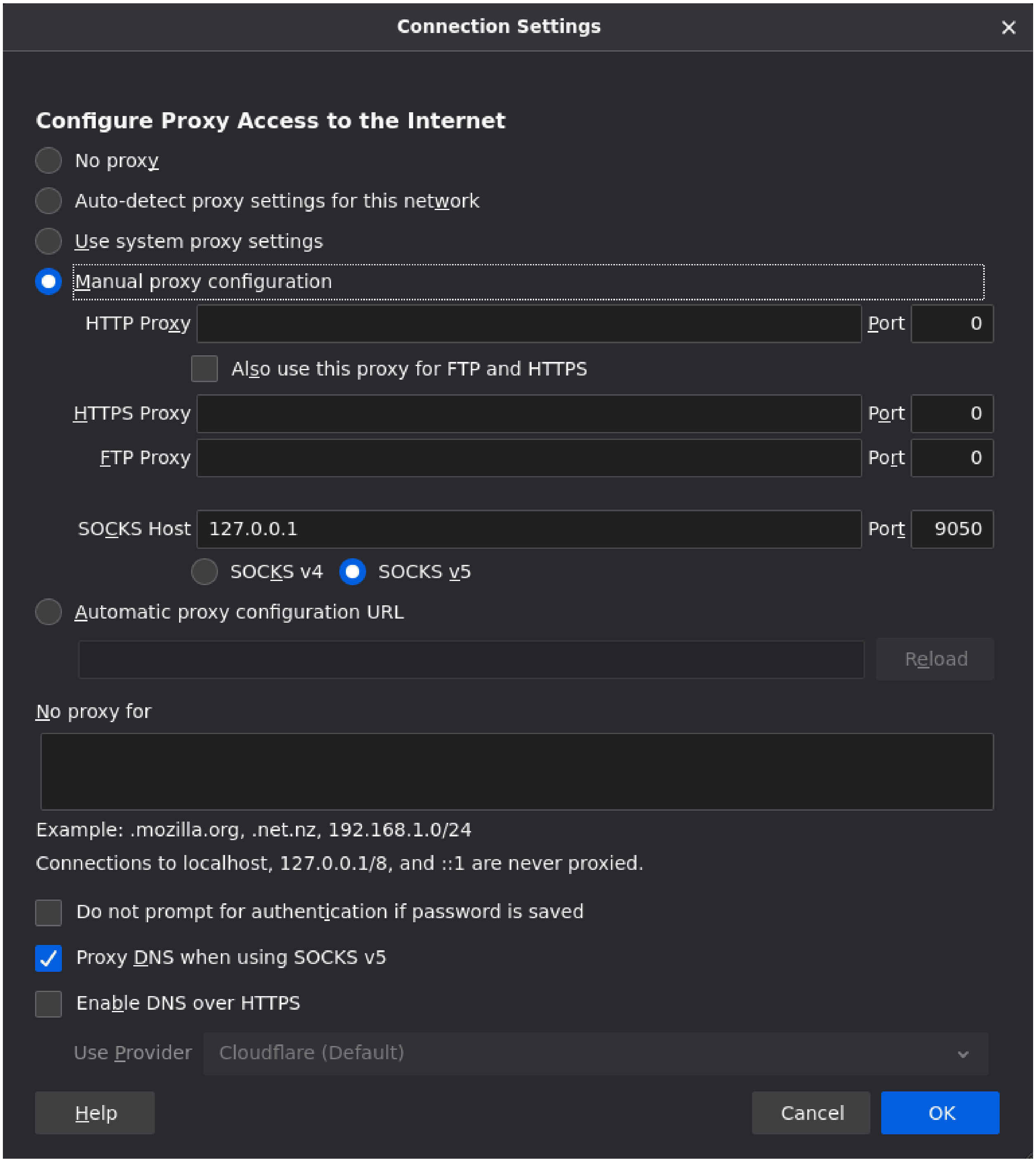
```
┌─[✗]─[glenn@quagmire]─[~]
└──╼ $tor
Apr 07 02:17:17.246 [notice] Tor 0.4.6.1-alpha-dev (git-5ebf2b81a1db144a) running on Linux with Libever
 2.1.11-stable, OpenSSL 1.1.1f, Zlib 1.2.11, Liblzma N/A, Libzstd N/A and Glibc 2.31 as libc.
Apr 07 02:17:17.246 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://w
.torproject.org/download/download#warning
Apr 07 02:17:17.246 [notice] This version is not a stable Tor release. Expect more bugs than usual.
Apr 07 02:17:17.246 [notice] Configuration file "/usr/local/etc/tor/torrc" not present, using reasonab
 defaults.
Apr 07 02:17:17.252 [notice] Opening Socks listener on 127.0.0.1:9050
Apr 07 02:17:17.252 [notice] Opened Socks listener connection (ready) on 127.0.0.1:9050
Apr 07 02:17:17.000 [notice] Parsing GEOIP IPv4 file /usr/local/share/tor/geoip.
Apr 07 02:17:17.000 [notice] Parsing GEOIP IPv6 file /usr/local/share/tor/geoip6.
Apr 07 02:17:17.000 [notice] Bootstrapped 0% (starting): Starting
Apr 07 02:17:17.000 [notice] Starting with guard context "default"
Apr 07 02:17:18.000 [notice] Bootstrapped 5% (conn): Connecting to a relay
Apr 07 02:17:18.000 [notice] Bootstrapped 10% (conn_done): Connected to a relay
Apr 07 02:17:18.000 [notice] Bootstrapped 14% (handshake): Handshaking with a relay
```

    - Verify that the tor program is listening on port 9050



```
┌─[glenn@quagmire]─[~]
└──╼ $sudo netstat -ltnp | grep "tor"
tcp        0      0 127.0.0.1:9050          0.0.0.0:*               LISTEN      1541794/tor
┌─[glenn@quagmire]─[~]
└──╼ $
```

    - Configure the proxy setting on firefox browser



Configure Proxy Access to the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy | Port 0

□ Also use this proxy for FTP and HTTPS

HTTPS Proxy | Port 0

FTP Proxy | Port 0

SOCKS Host 127.0.0.1 | Port 9050

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

□ Do not prompt for authentication if password is saved

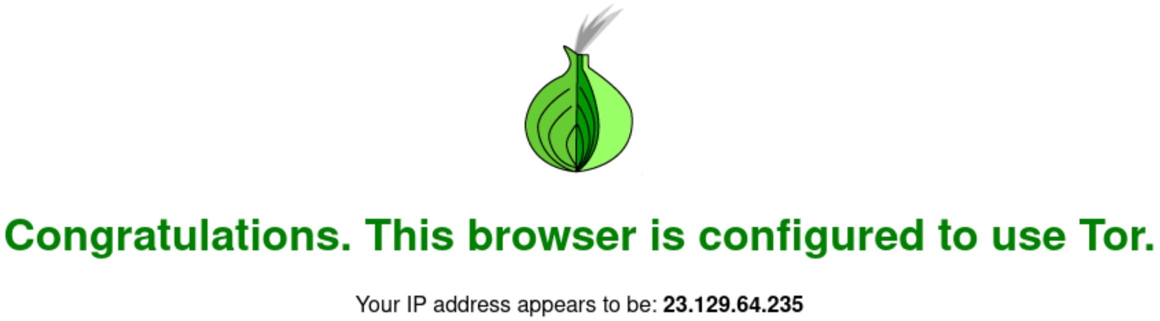
☑ Proxy DNS when using SOCKS v5

□ Enable DNS over HTTPS

Use Provider Cloudflare (Default)

- Confirm that the web traffic is relaying via tor circuit,

  Visit the website: https://check.torproject.org/ for confirmation



Congratulations. This browser is configured to use Tor.

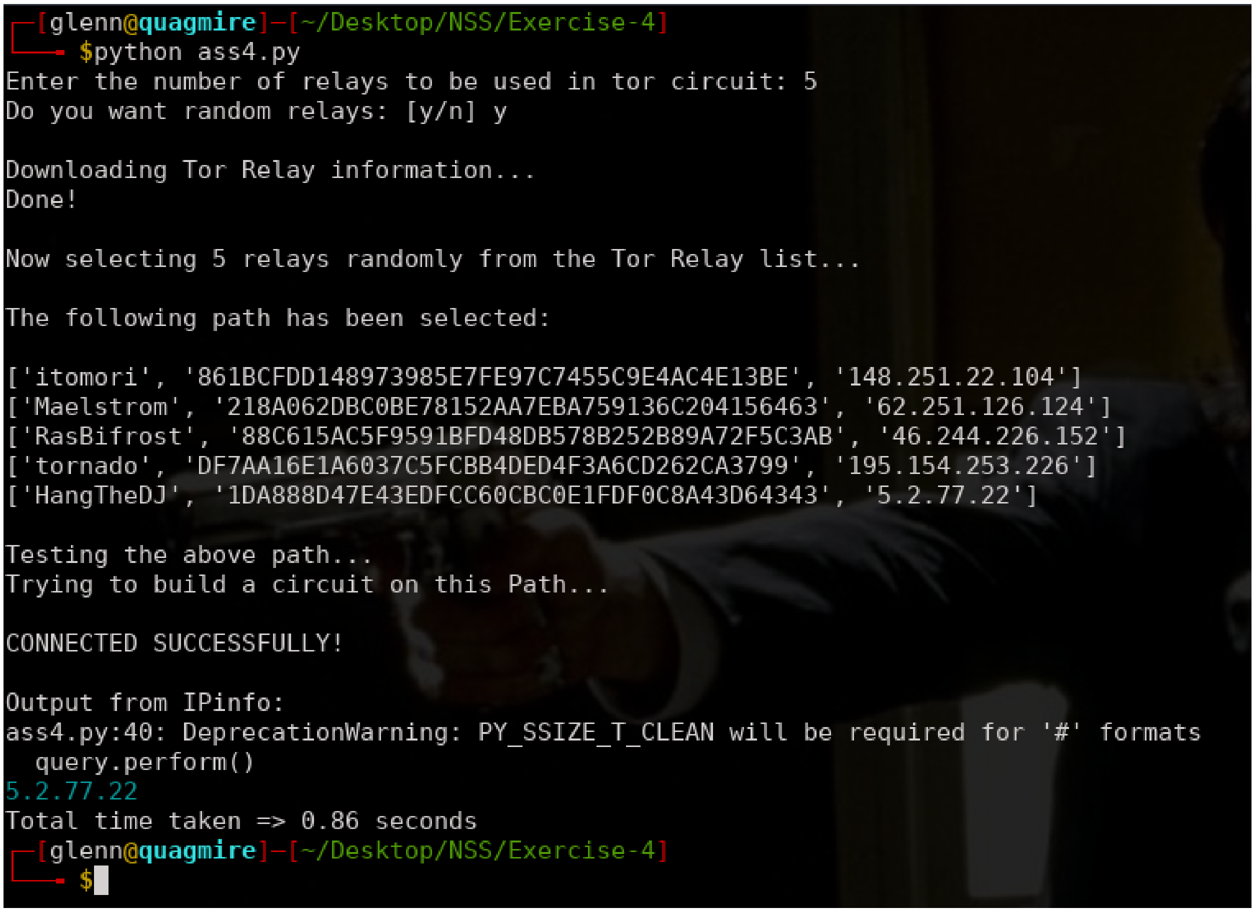
Your IP address appears to be: **23.129.64.235**

As we can see, the configurations are correct

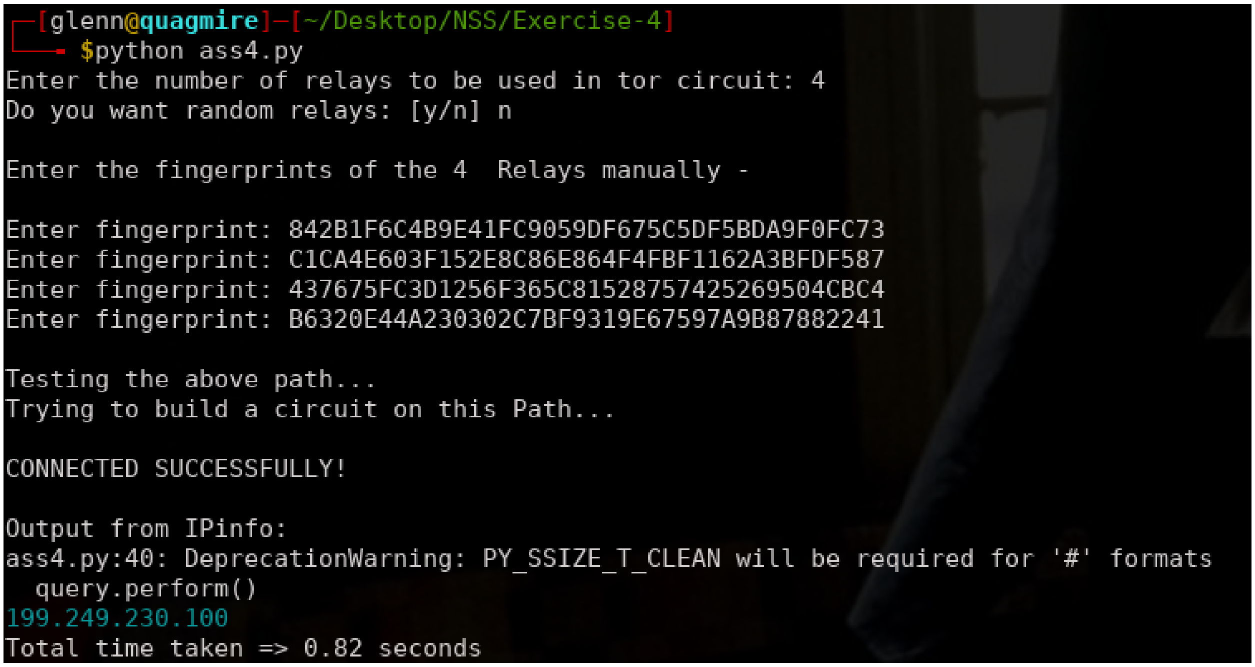
STEP 2 : Using python scripts (stem)

- Install the stem library: pip3 install stem
- Copy the torrc file from /usr/local/etc/tor/torrc.sample to the home folder and rename it .torrc
- Disable the following comments in this .torrc file:
  - ControlPort 9051
  - CookieAuthentication 1
- Writing a python script using stem library to control tor -
  - The script has been attached in the zip
  - It takes number of relays as input
  - Then, either the relays can be selected randomly from the Tor Relay List, or can configured manually from their fingerprints

- Screenshots for both ways have been attached below:



```
┌─[glenn@quagmire]─[~/Desktop/NSS/Exercise-4]
└──╼ $python ass4.py
Enter the number of relays to be used in tor circuit: 5
Do you want random relays: [y/n] y

Downloading Tor Relay information...
Done!

Now selecting 5 relays randomly from the Tor Relay list...

The following path has been selected:

['itomori', '861BCFDD148973985E7FE97C7455C9E4AC4E13BE', '148.251.22.104']
['Maelstrom', '218A062DBC0BE78152AA7EBA759136C204156463', '62.251.126.124']
['RasBifrost', '88C615AC5F9591BFD48DB578B252B89A72F5C3AB', '46.244.226.152']
['tornado', 'DF7AA16E1A6037C5FCBB4DED4F3A6CD262CA3799', '195.154.253.226']
['HangTheDJ', '1DA888D47E43EDFCC60CBC0E1FDF0C8A43D64343', '5.2.77.22']

Testing the above path...
Trying to build a circuit on this Path...

CONNECTED SUCCESSFULLY!

Output from IPinfo:
ass4.py:40: DeprecationWarning: PY_SSIZE_T_CLEAN will be required for '#' formats
  query.perform()
5.2.77.22
Total time taken => 0.86 seconds
┌─[glenn@quagmire]─[~/Desktop/NSS/Exercise-4]
└──╼ $
```



```
┌─[glenn@quagmire]─[~/Desktop/NSS/Exercise-4]
└──╼ $python ass4.py
Enter the number of relays to be used in tor circuit: 4
Do you want random relays: [y/n] n

Enter the fingerprints of the 4  Relays manually -

Enter fingerprint: 842B1F6C4B9E41FC9059DF675C5DF5BDA9F0FC73
Enter fingerprint: C1CA4E603F152E8C86E864F4FBF1162A3BFDF587
Enter fingerprint: 437675FC3D1256F365C815287574252695O4CBC4
Enter fingerprint: B6320E44A230302C7BF9319E67597A9B87882241

Testing the above path...
Trying to build a circuit on this Path...

CONNECTED SUCCESSFULLY!

Output from IPinfo:
ass4.py:40: DeprecationWarning: PY_SSIZE_T_CLEAN will be required for '#' formats
  query.perform()
199.249.230.100
Total time taken => 0.82 seconds
```

As we can see, we get the new public IP from the TOR exit node, which can be verified with the last relay fingerprint.