

1)

Install necessary libraries:

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

Install the tor repo and cd into it

```
git clone https://git.torproject.org/tor.git
```

Run autogen.sh to generate all initial makefiles:

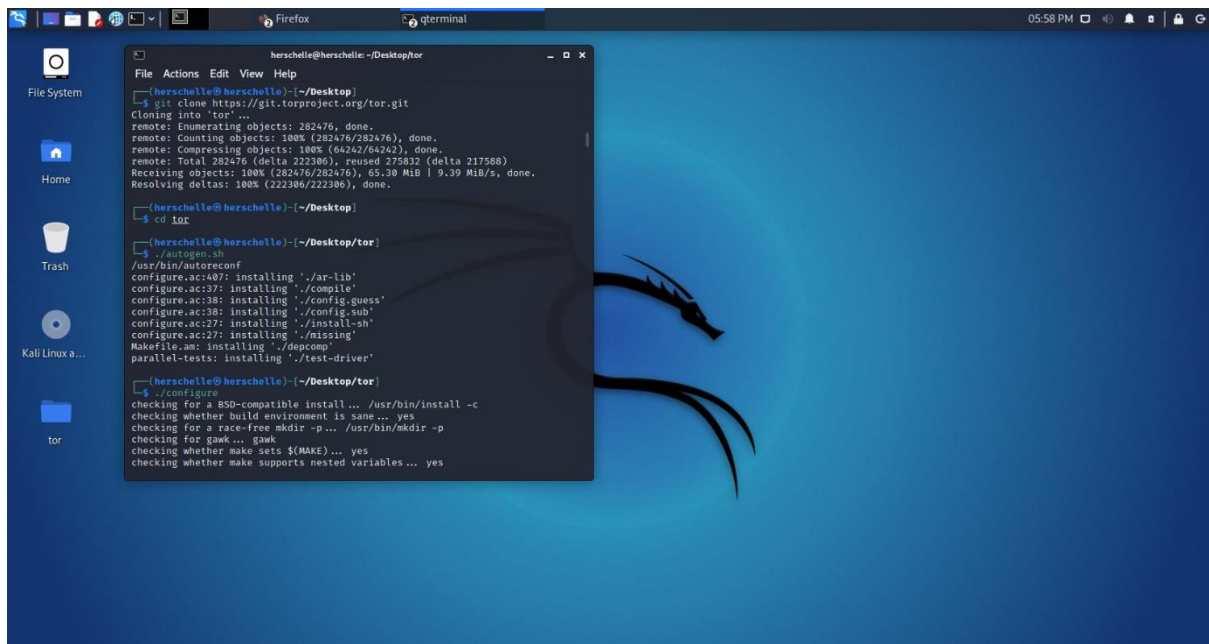
```
./autogen.sh
```

Prepare tor to be built:

```
./configure --disable-asciidoc # do not build manpages
```

Compile and install locally:

```
sudo make && make install
```

A screenshot of a Kali Linux desktop environment. The background is a blue wallpaper with a dragon logo. On the left is a sidebar with icons for File System, Home, Trash, and Kali Linux a... Below these is a folder icon labeled 'tor'. In the center is a terminal window titled 'herschelle@herschelle: ~/Desktop/tor'. The terminal shows the following commands and output:

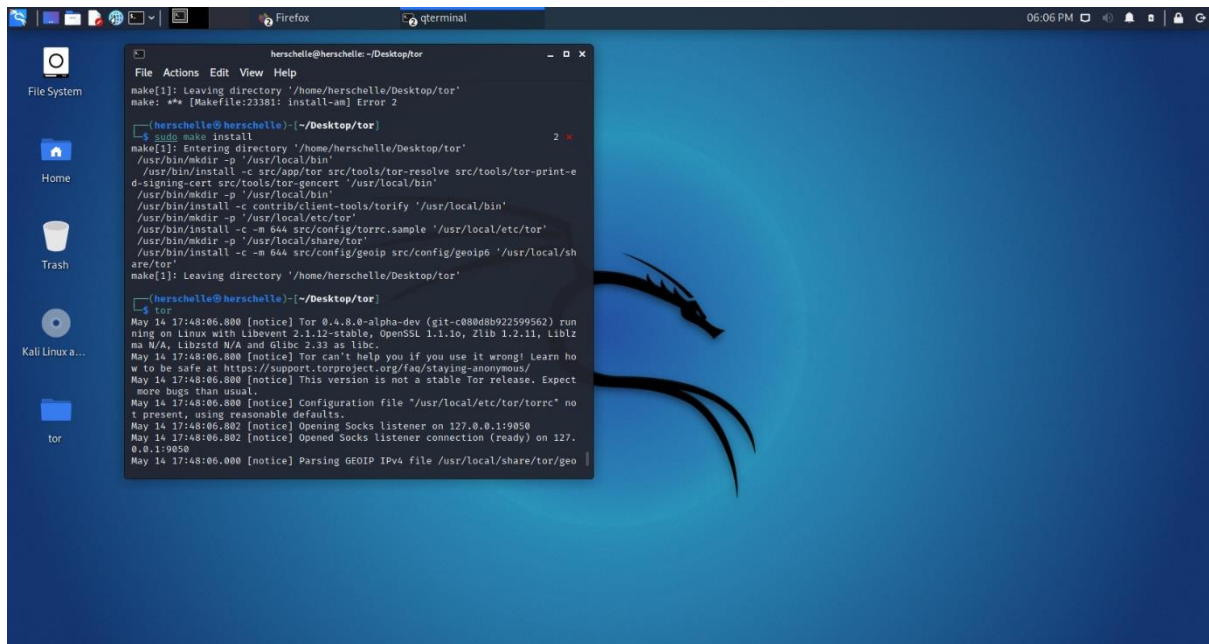
```
(herschelle@herschelle):~/Desktop
$ git clone https://git.torproject.org/tor.git
Cloning into 'tor' ...
remote: Enumerating objects: 282476, done.
remote: Counting objects: 100% (282476/282476), done.
remote: Compressing objects: 100% (64262/64262), done.
remote: Total 282476 (delta 222386), reused 275832 (delta 217568)
Receiving objects: 100% (282476/282476), 65.30 MiB | 9.39 MiB/s, done.
Resolving deltas: 100% (222386/222386), done.

(herschelle@herschelle):~/Desktop
$ cd tor

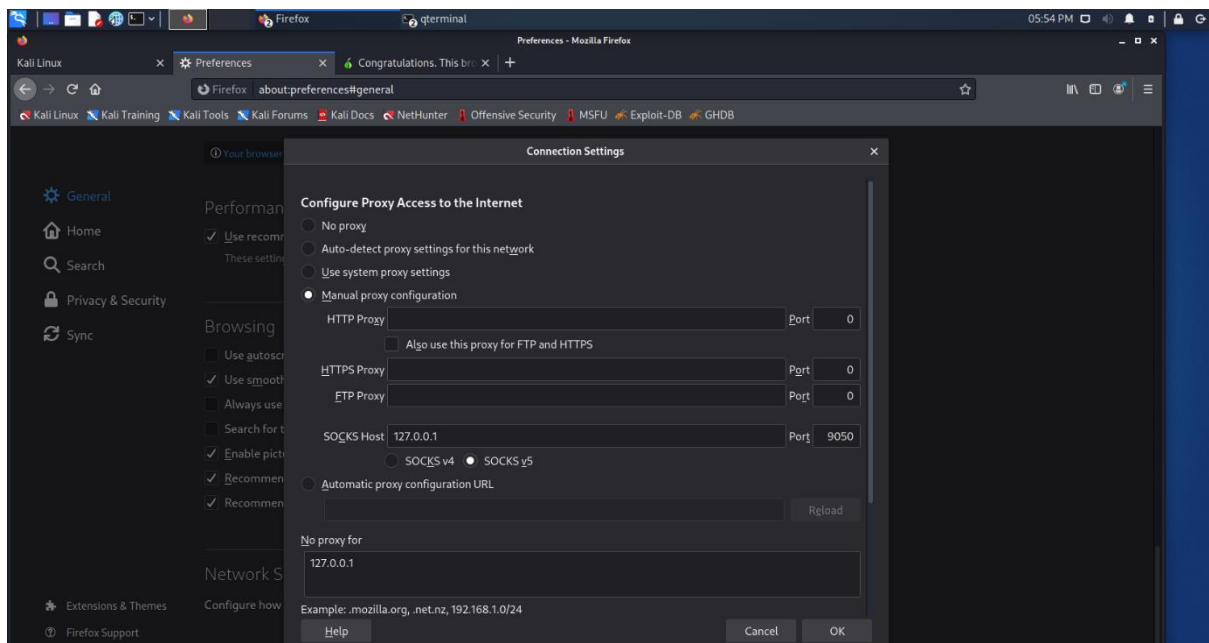
(herschelle@herschelle):~/Desktop/tor
$ ./autogen.sh
/usr/bin/autoreconf
configure.ac:40: installing './ar-lib'
configure.ac:37: installing './compile'
configure.ac:38: installing './config.guess'
configure.ac:38: installing './config.sub'
configure.ac:27: installing './install-sh'
configure.ac:27: installing './missing'
Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'

(herschelle@herschelle):~/Desktop/tor
$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
```

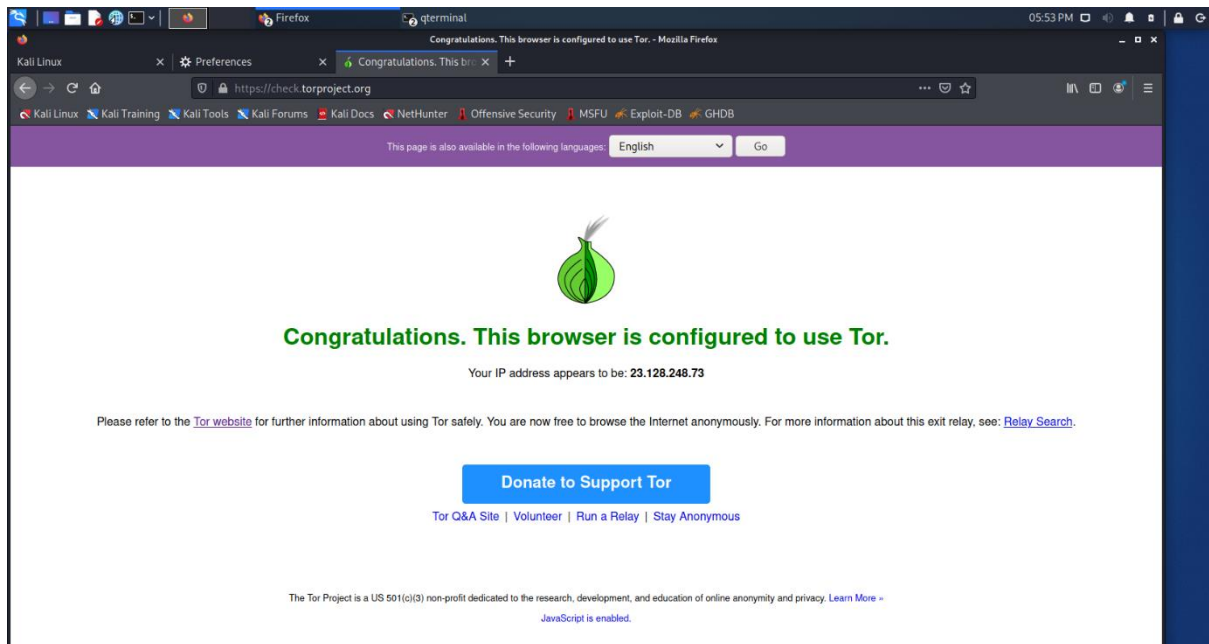
We can run Tor simply typing “tor” in the terminal



Finally configure the browser as shown below.

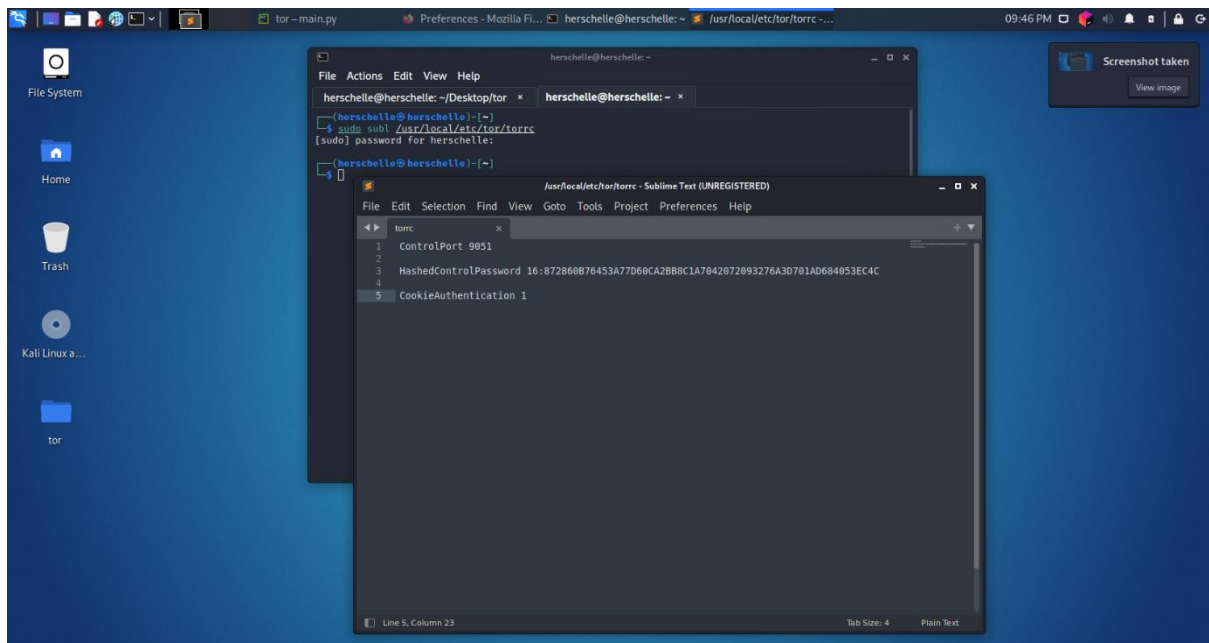


We can check if the browser is successfully configured to use Tor, go to the website [check.torproject.org](http://check.torproject.org)

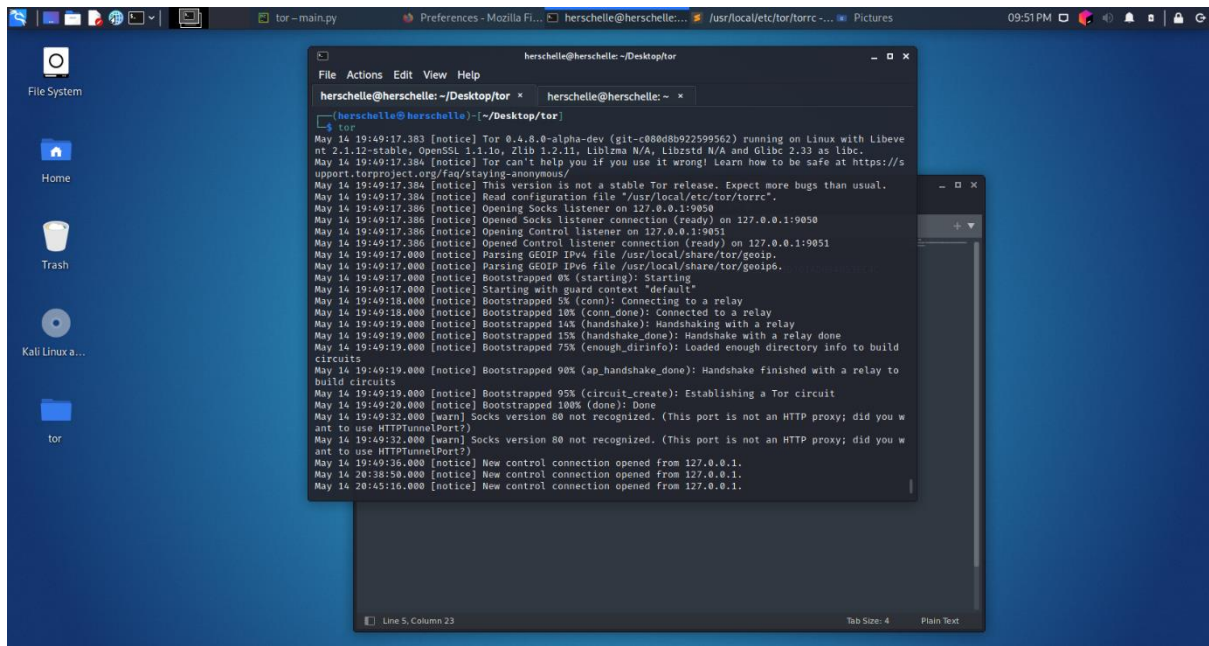


Now we create a file in `/usr/local/etc/` directory named “torrc” and set the contents as shown below.

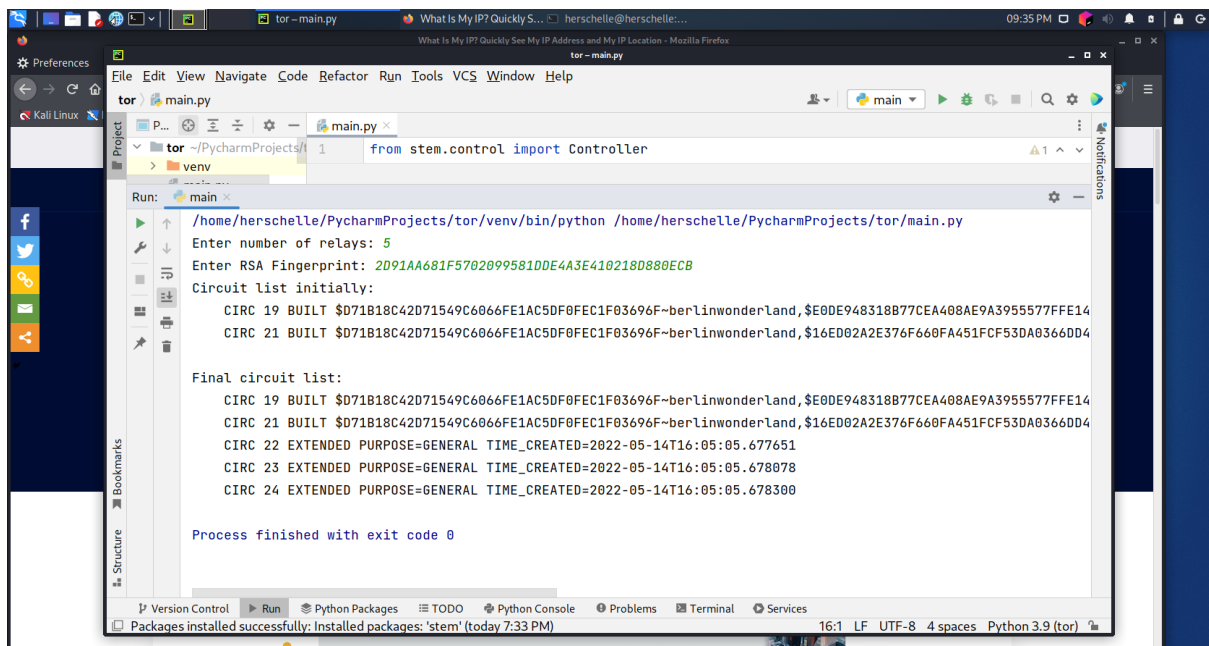
[https://stem.torproject.org/tutorials/the\\_little\\_relay\\_that\\_could.html](https://stem.torproject.org/tutorials/the_little_relay_that_could.html)



Start the tor program in the terminal typing “tor”



Finally run the python file “part1.py”:

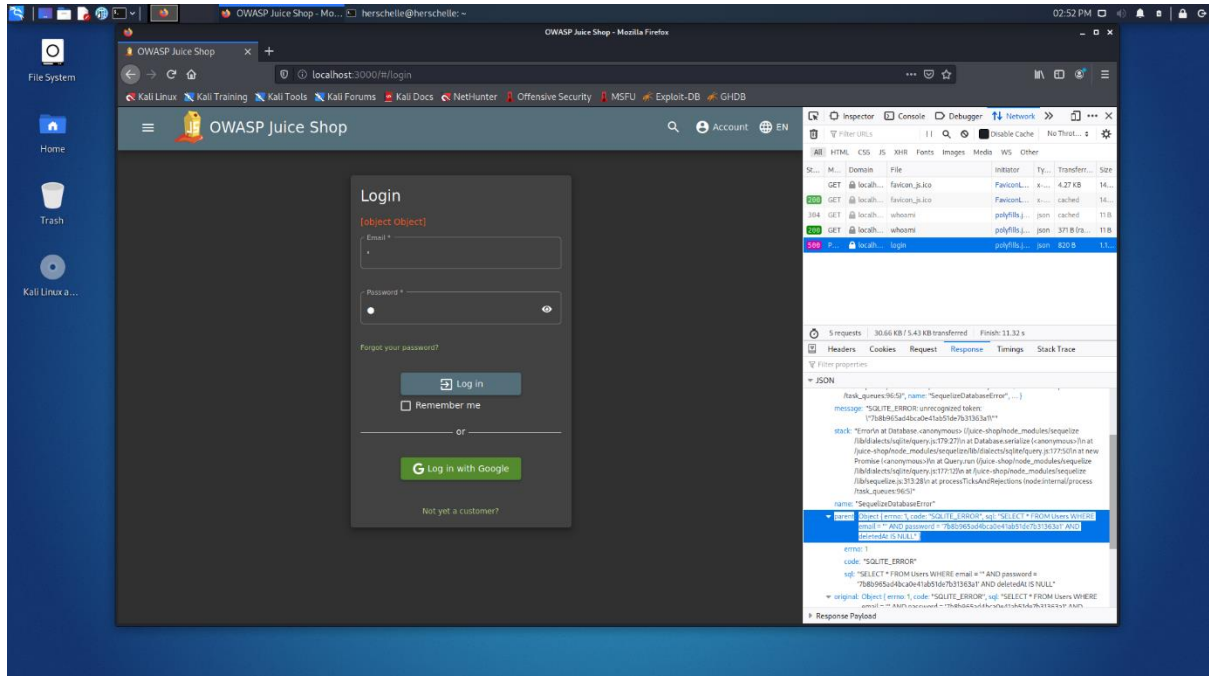


Initially, 2 circuits were present and 3 more were added for input 5.

3)

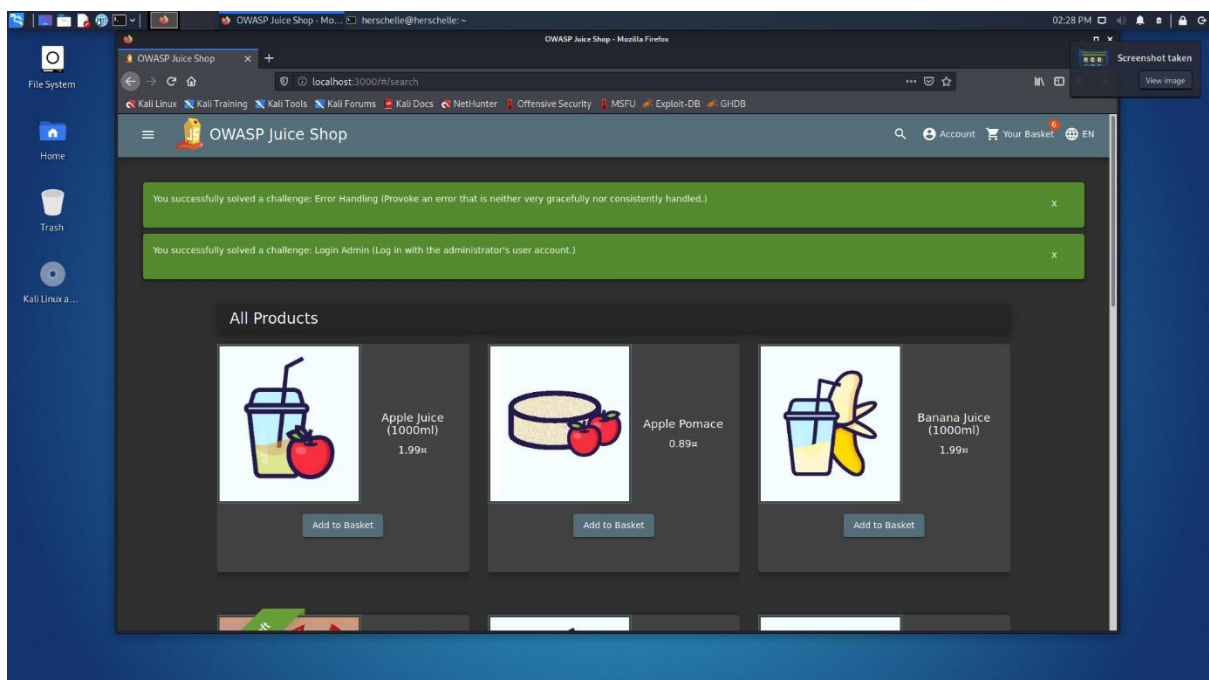
a)

By typing '(single quote) in email and anything is passwords, we see that email field is vulnerable. If we go to the network tab, we can see the actual code being using to fetch from the database.



We send ' OR TRUE -- in the email

Now the query will check for email = " or TRUE, and "-- will comment out the rest of the query ahead of it.

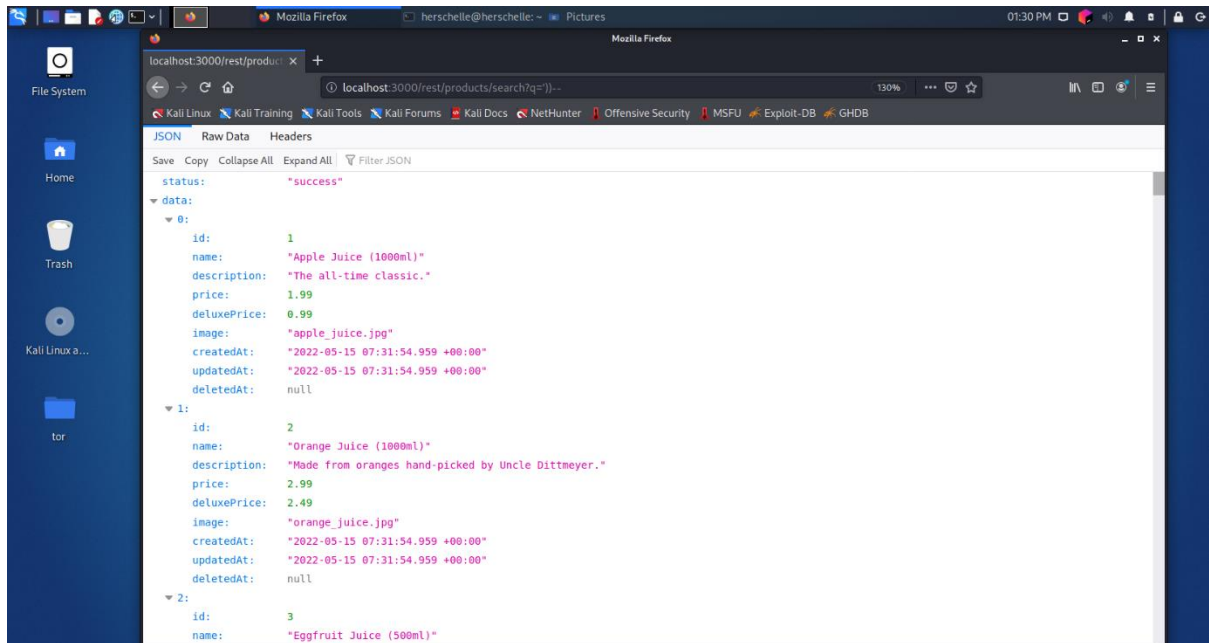


b)

Append `"/rest/products/search?q="` to url

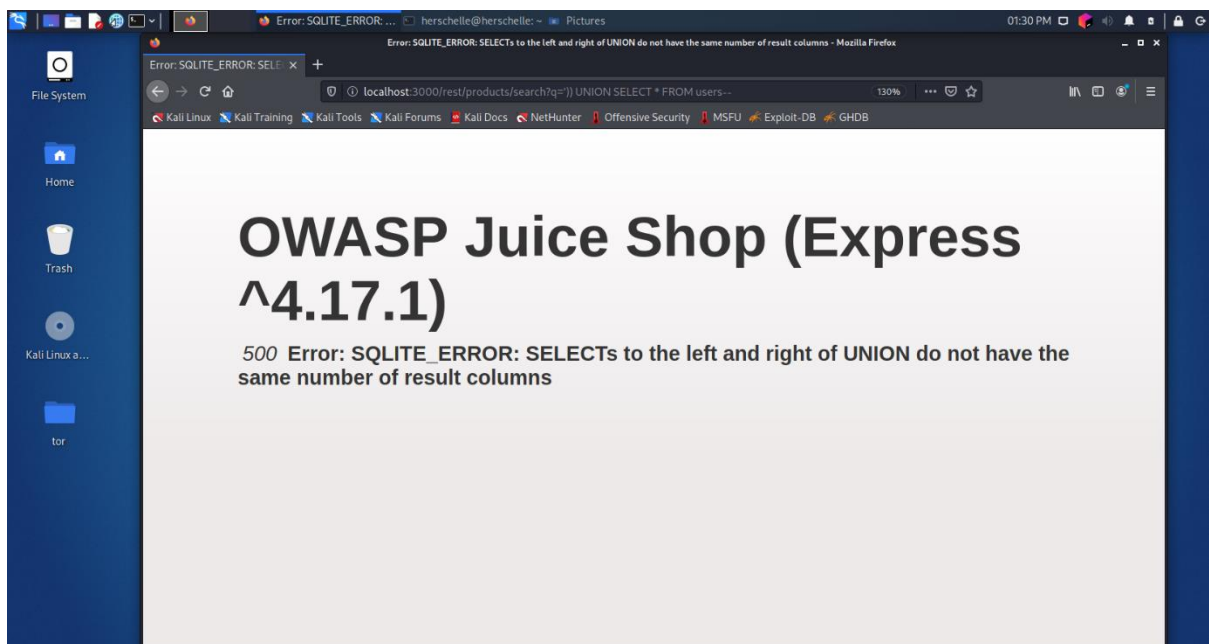
and query: `'))--`

We see this is vulnerable to injection.



Now by trying union statement as `'))-- union select * from users --`

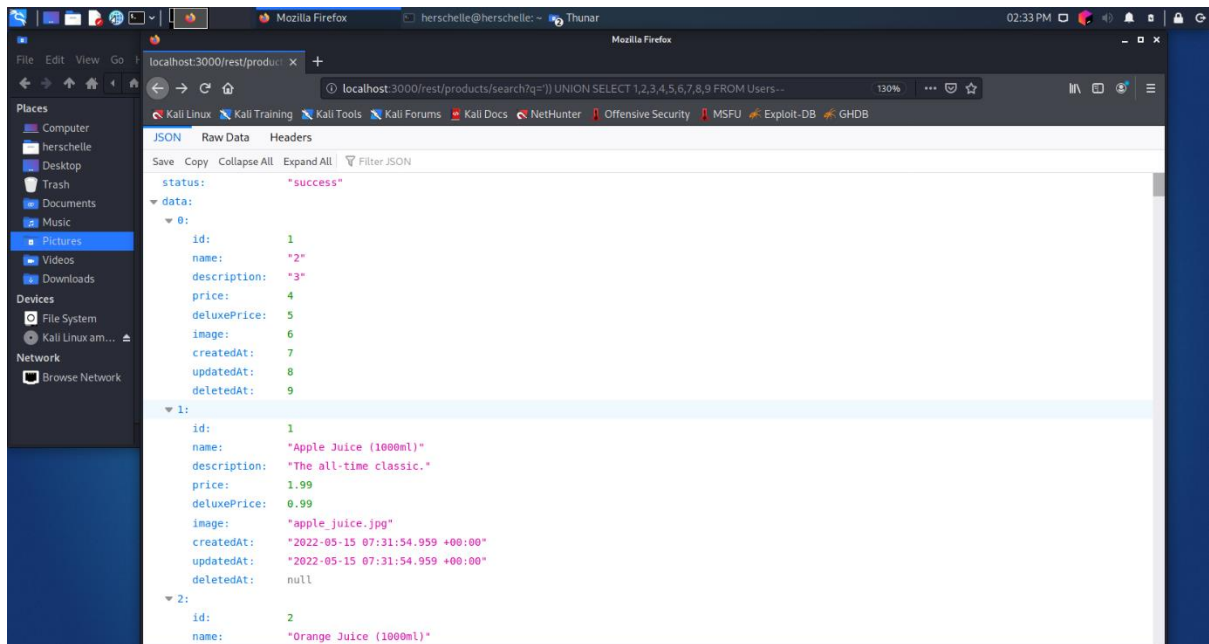
We see the following error



By trial and error, 9 values work.

Query: `')) UNION SELECT 1,2,3,4,5,6,7,8,9 FROM Users--`





Change the numbers to table column names

Query: `')) UNION SELECT id, username, password, email,5,6,7,8,9 FROM Users—`

