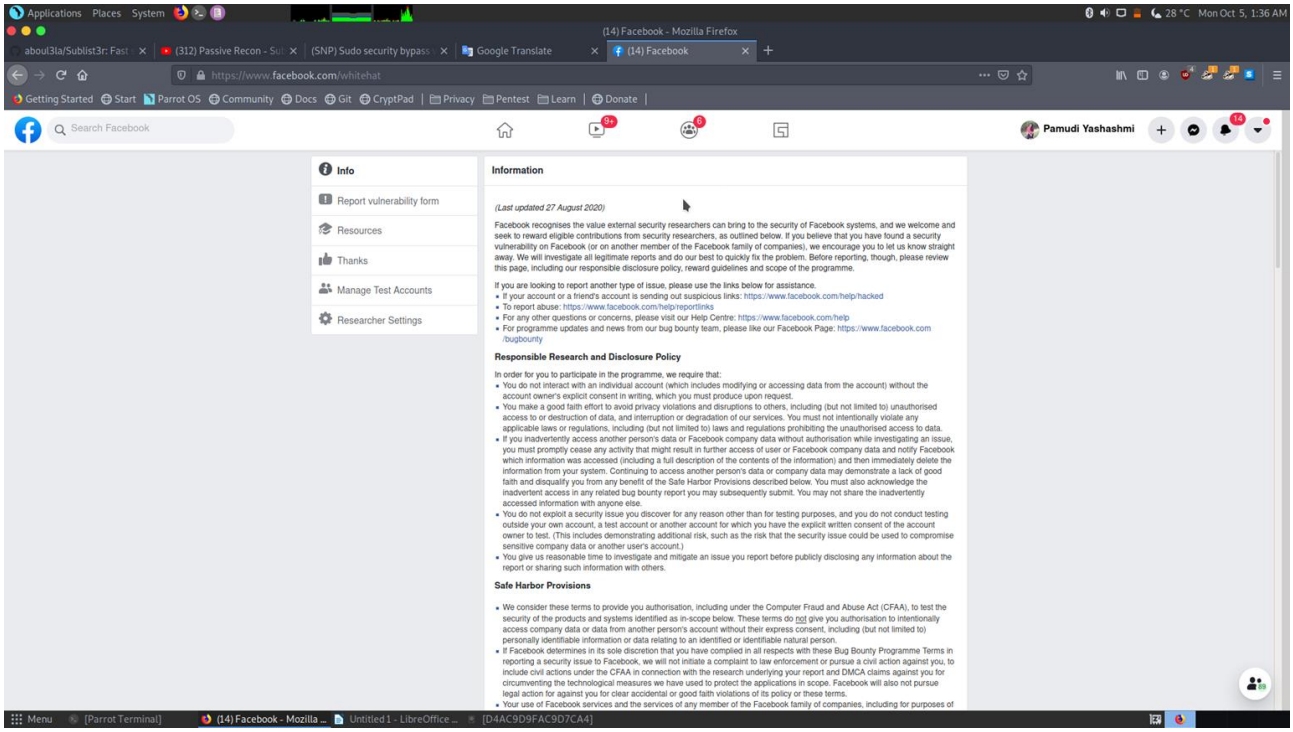
Penetration Testing on Facebook Domains

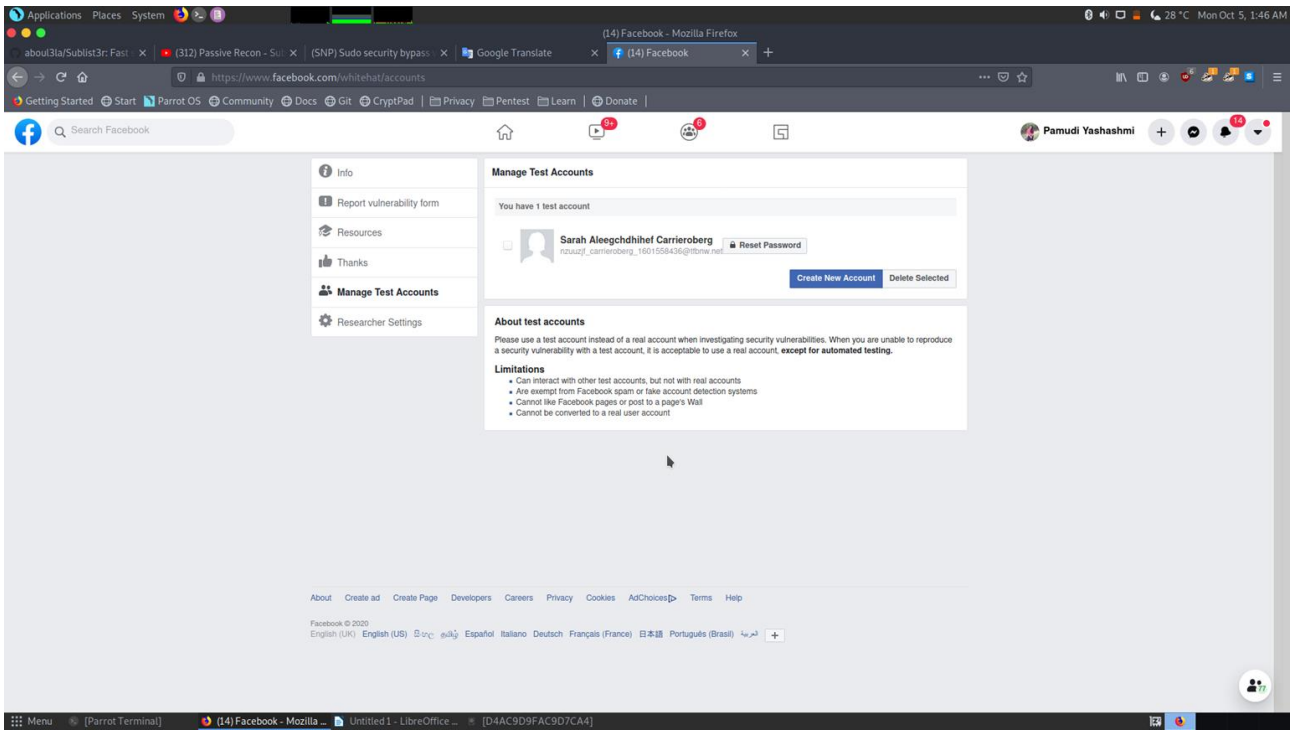
Pamudi Meeriyagalla

Methodology

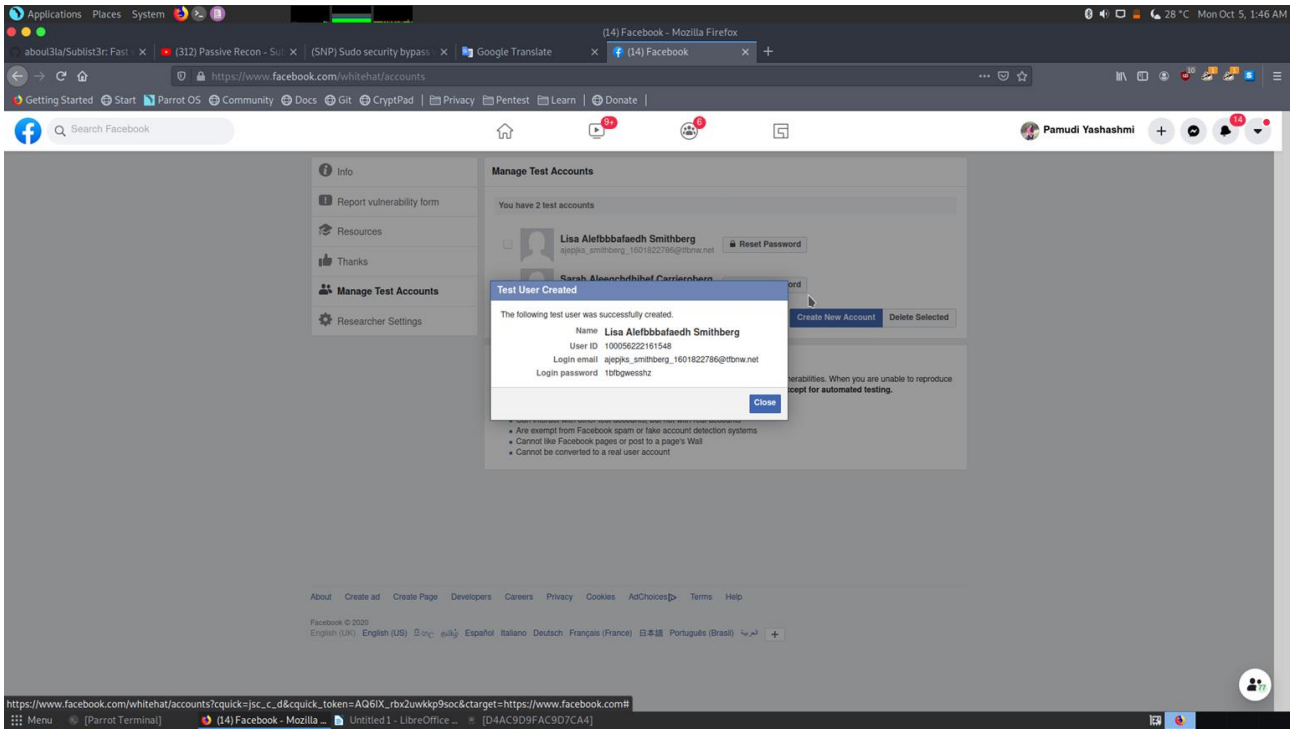
The domain which I selected for the Web Security assignment is facebook.com. Facebook is a social media website that include very sensitive information. Before selected that domain I read the privacy policy of that domain (https://www.facebook.com/whitehat). To join to the whitehat program of the facebook needed to create a test account. According to the privacy policy sheet of the facebook, without creating a test account, it is not legal to use automated tools.



To create the test account need to go to the “Manage Test Accounts” and click on create test account button on the web page.



Using test accounts can not interact with real accounts but it can use to interact with other test accounts. Test accounts are exempt from Facebook spam or fake account detection systems. Using test accounts can not like facebook pages or post to a page's Wall. Test accounts can not convert to a real accounts.



Using test accounts can do the penetration testings.

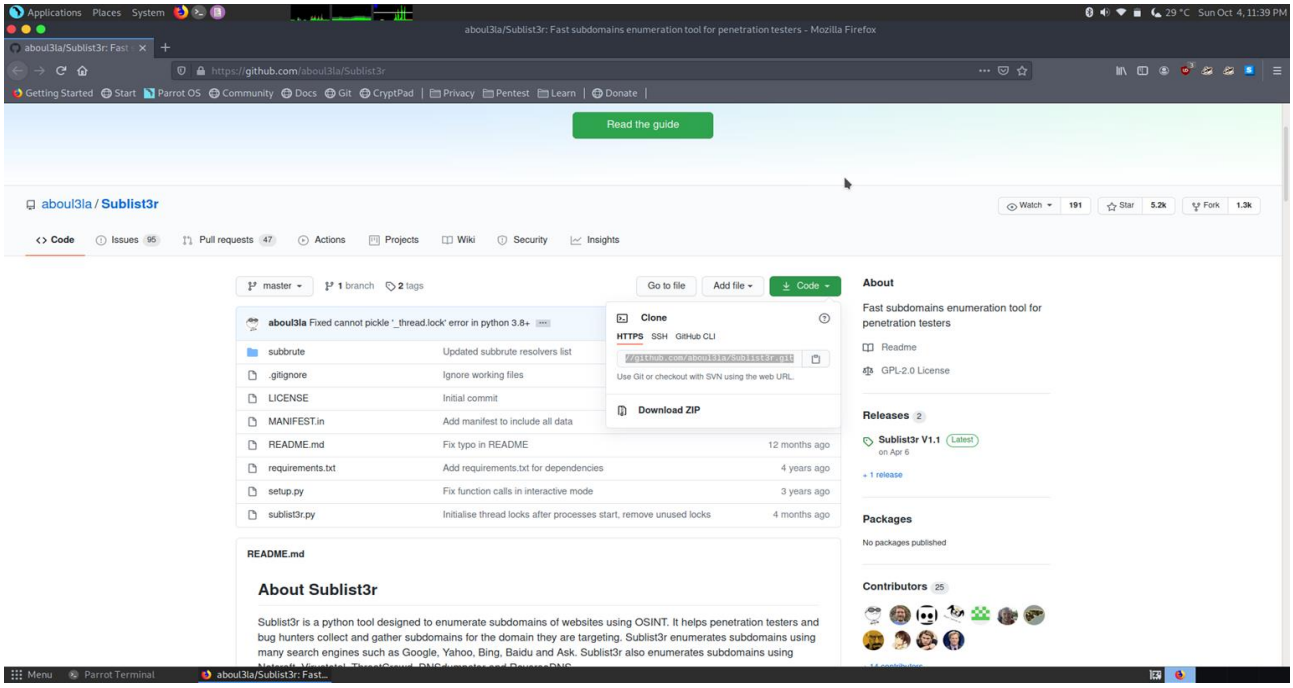
Auditing

Sublist3r

First of all had to find the number of subdomains of the selected domain. For that, used “Sublist3r”. This tool (Sublist3r) is created by using Python language. To use sublist3r, can use Python 2 and Python3. In here I used Python3 (pip3).

Step 01: Clone the repository file. To clone the file need to go to the github page and copy the link and after that in the terminal need to type,

#git clone https://github.com/aboul3la/Sublist3r.git (after ‘git clone’ paste the copied link)



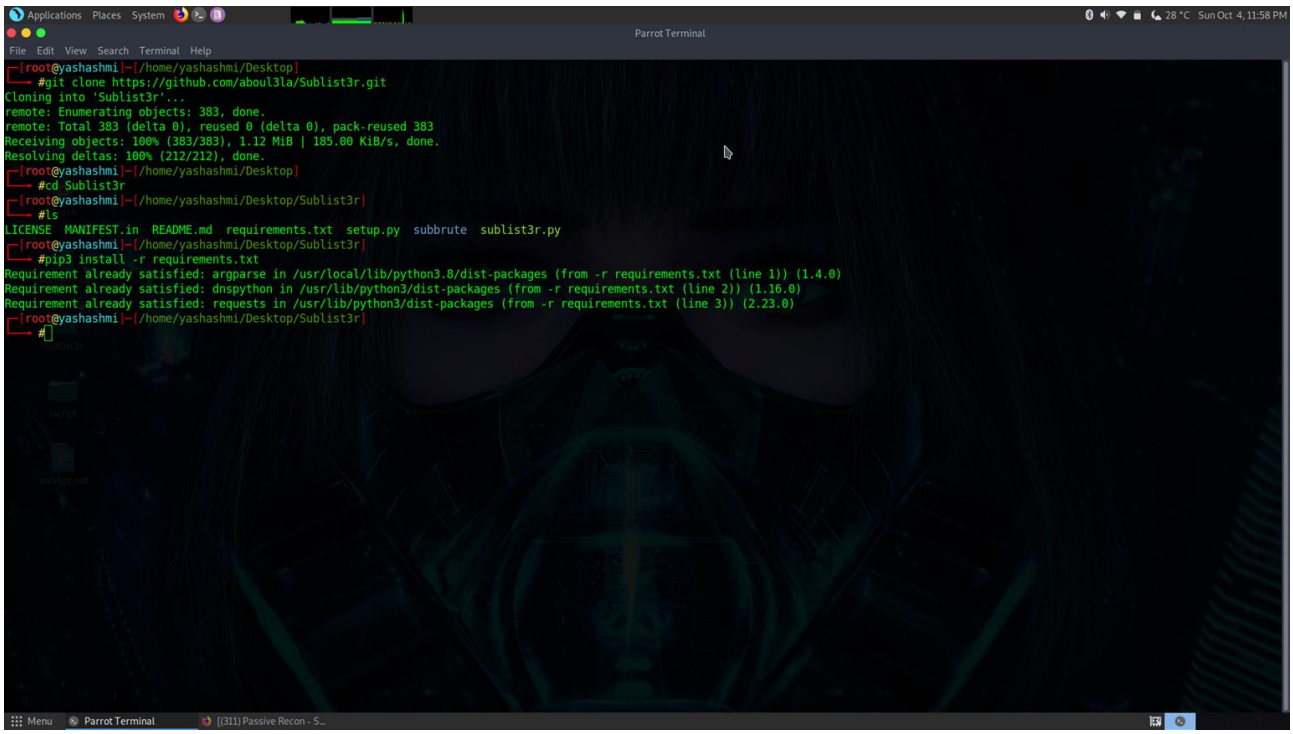
Step 02: After cloning the Sublist3r, go in to the Sublist3r directory file and list down the data using commands,

#ls

#cd Sublist3r

Step 03: To run the Sublist3r tool there are some requirements need. To fulfill those requirements install the requirements.txt file using a command,

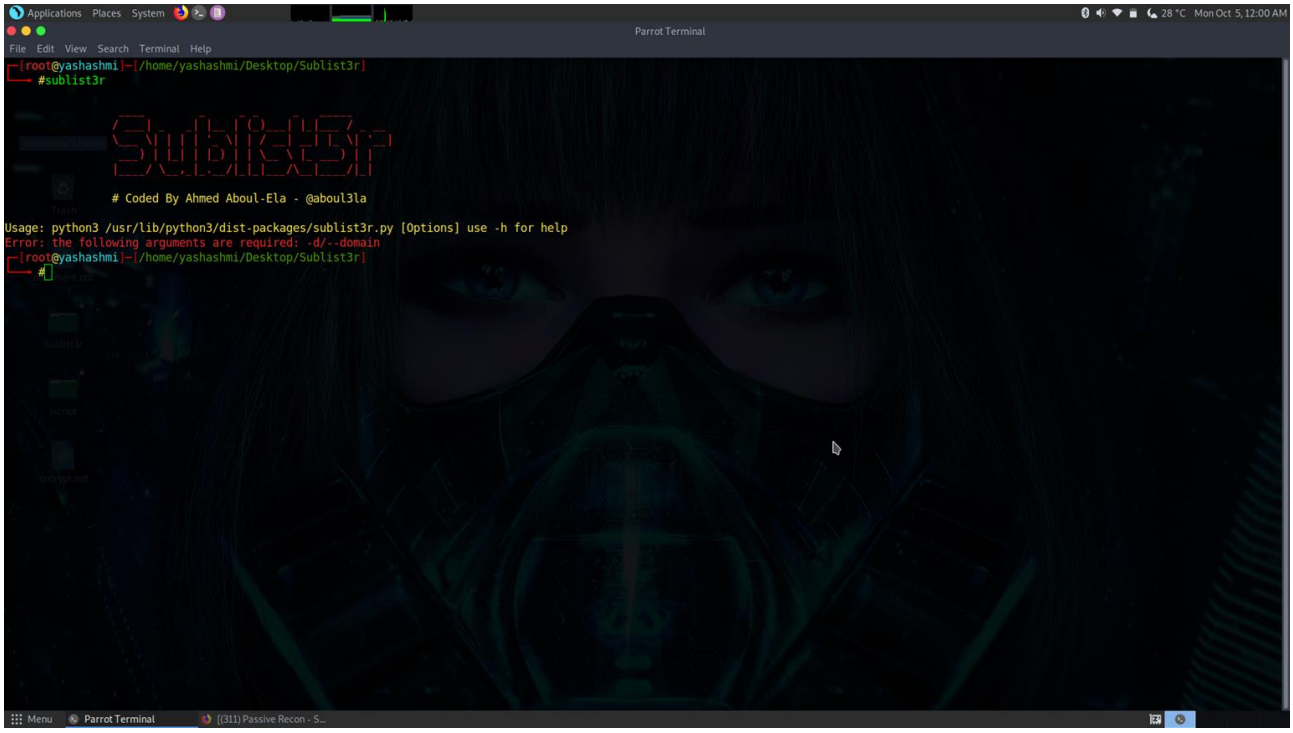
#pip3 install -r requirements.txt



Step 04: After successfully installing the requirements.txt file can run the tool. To use the tool give a command,

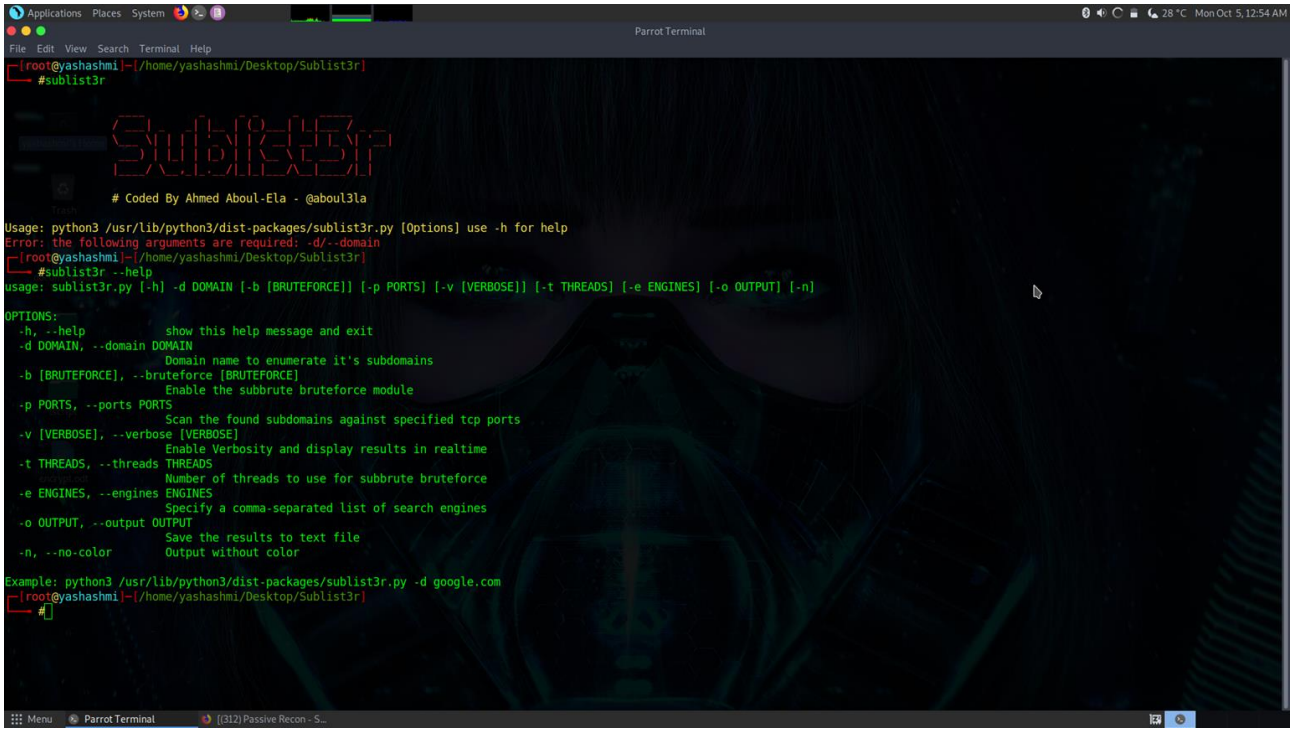
#sublist3r

After giving this command, can use the tool.



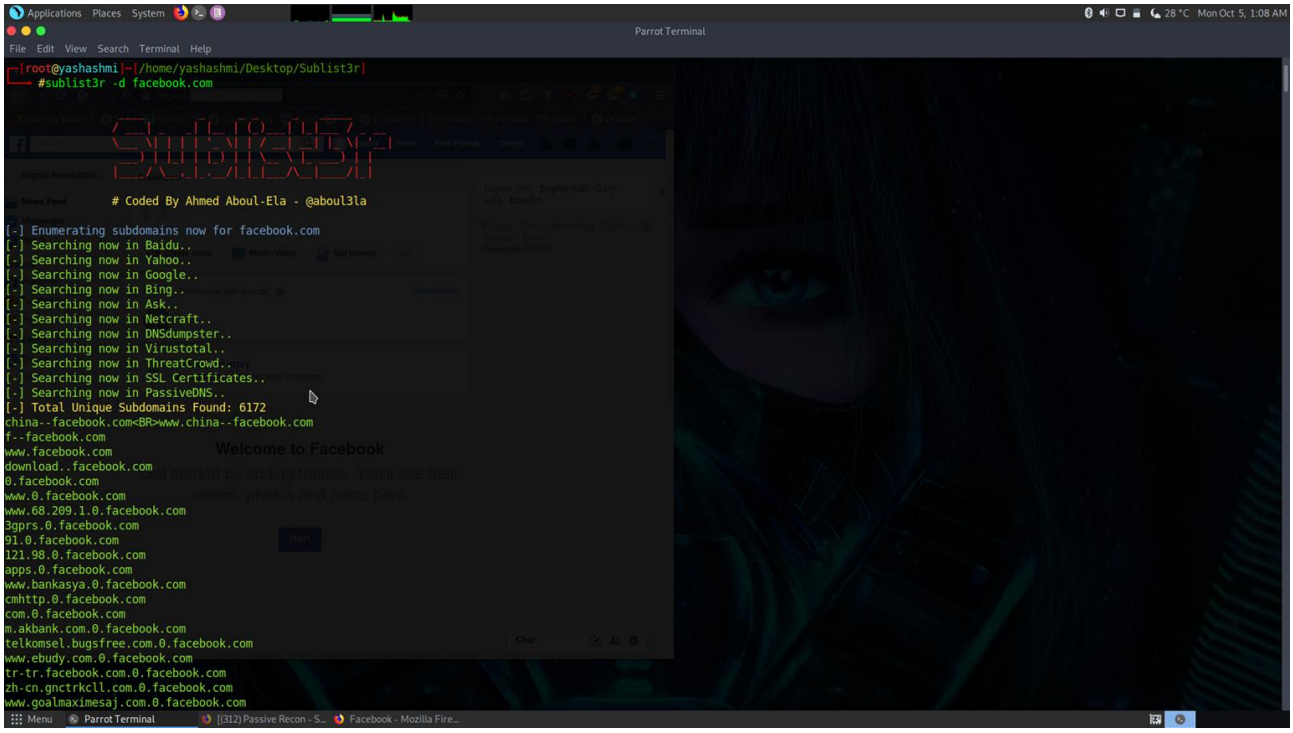
Step 05: There is a help menu in Sublist3r. By using it, can have a knowledge about how to use the tool properly. Command to display the help menu,

#sublist3r –help



-v: option for verbose. Verbose will show the subdomains in real-time.

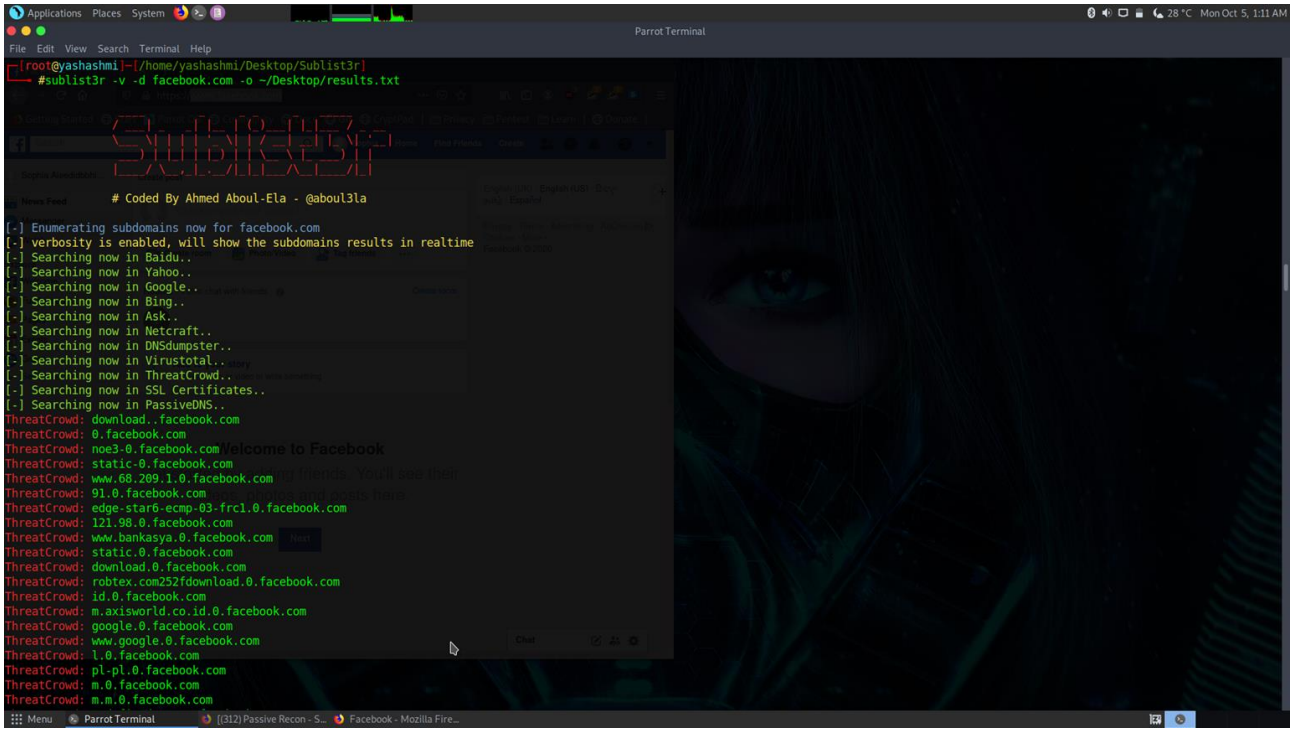
-d: Domain name to enumerate it’s subdomains. # sublist3r -d facebook.com

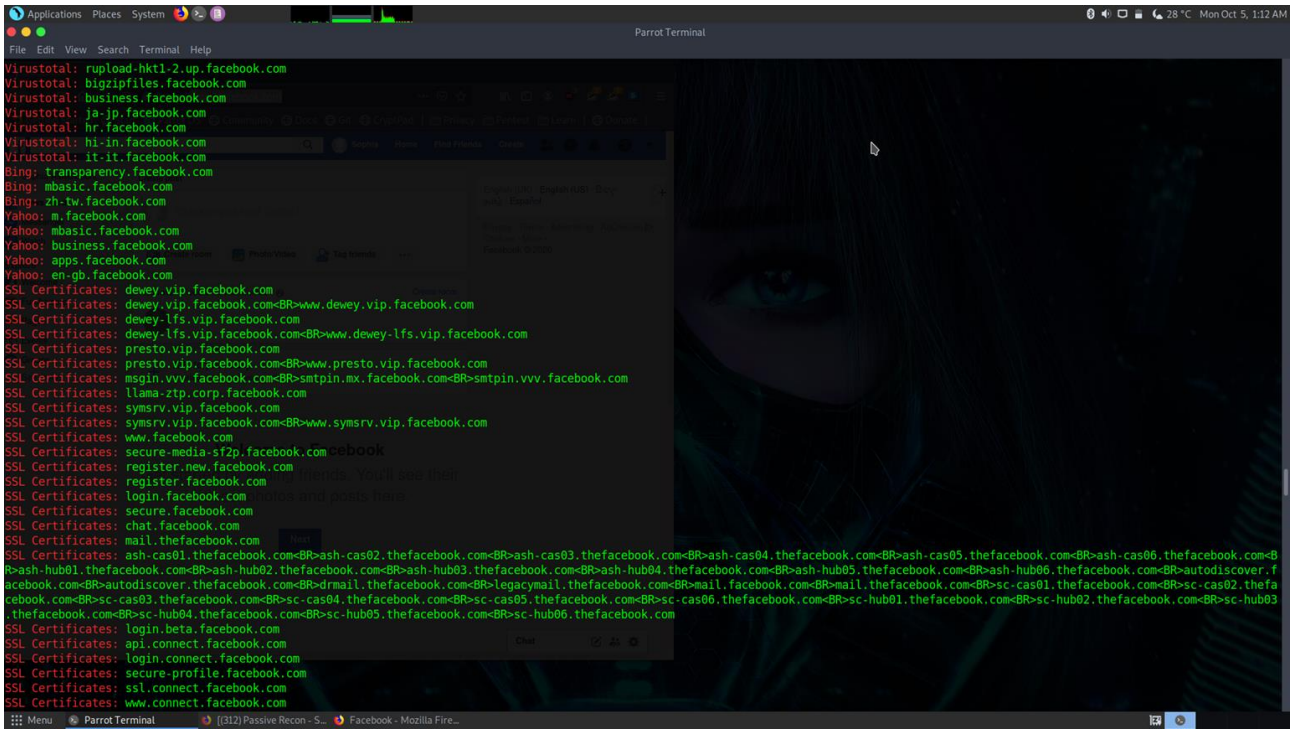


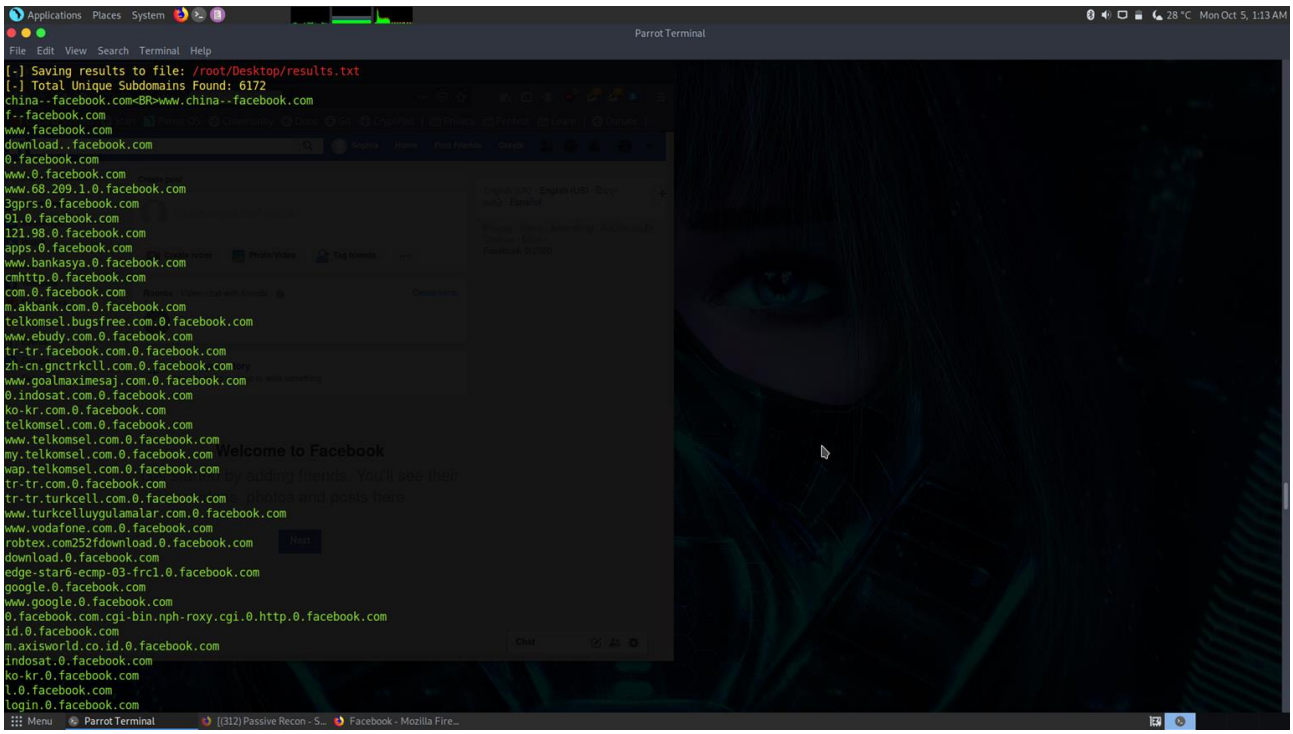
-o: save the out put in to a text file.

Command to use -v, -d, and -o together,

#sublist3r -v -d facebook.com -o ~/Desktop/results.txt (“~/Desktop/results.txt” this is the path to save the out put)

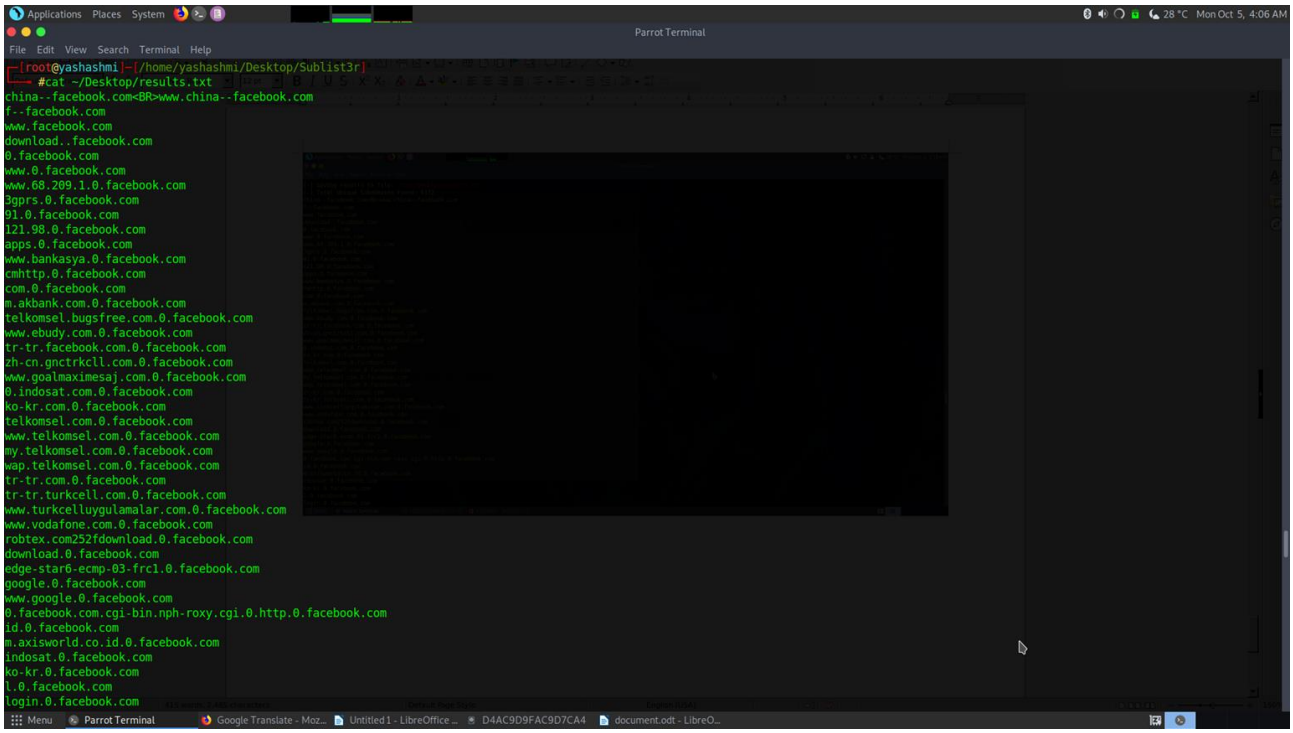






Using “cat” command can retrieve the data of the results.txt file.

#cat ~/Desktop/results.txt

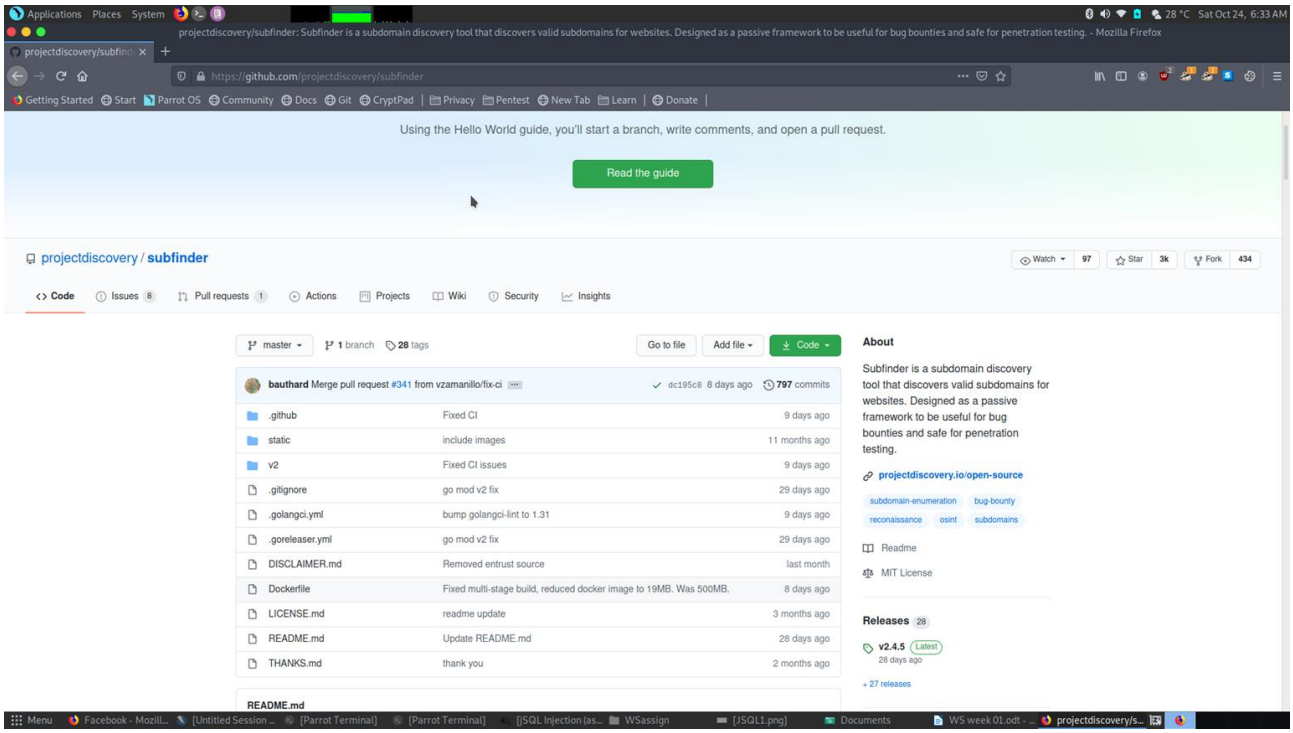


By using Sublist3r, found the number of subdomains of the facebook: 6172.

Subfinder

Firstly used Sublist3r to find the number of subdomains. After using that tool, again used a tool called “subfinder”. Used two tools to find the number of subdomains because to get a more accurate answer. “subfinder” is also a subdomain discovery tool that discovers valid subdomains for websites. This tool is designed as a passive framework and this is very useful for bug-bounty programs and for penetration testings. This tool is quicker than “Sublist3r”.

This one is also a git hub tool. To use this tool first of all need to download this and install this. To do that used below steps.



#git clone https://github.com/projectdiscovery/subfinder.git

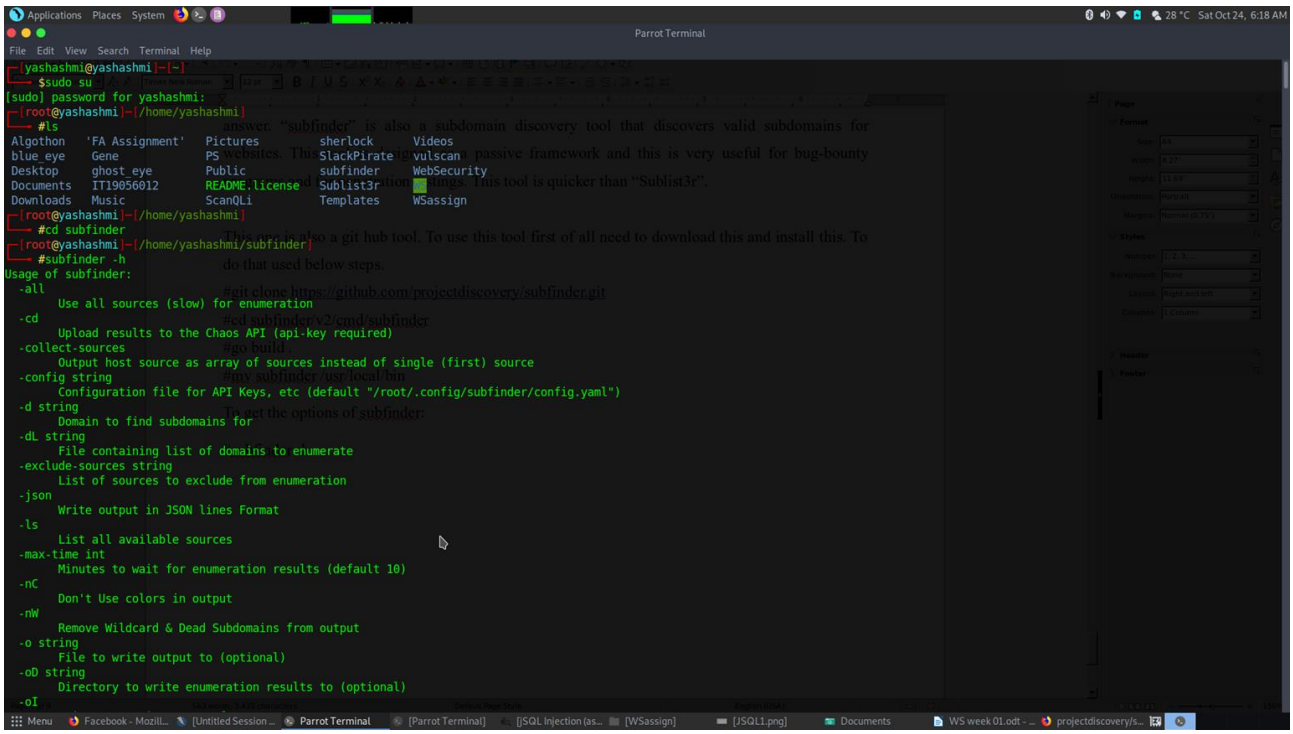
#cd subfinder/v2/cmd/subfinder

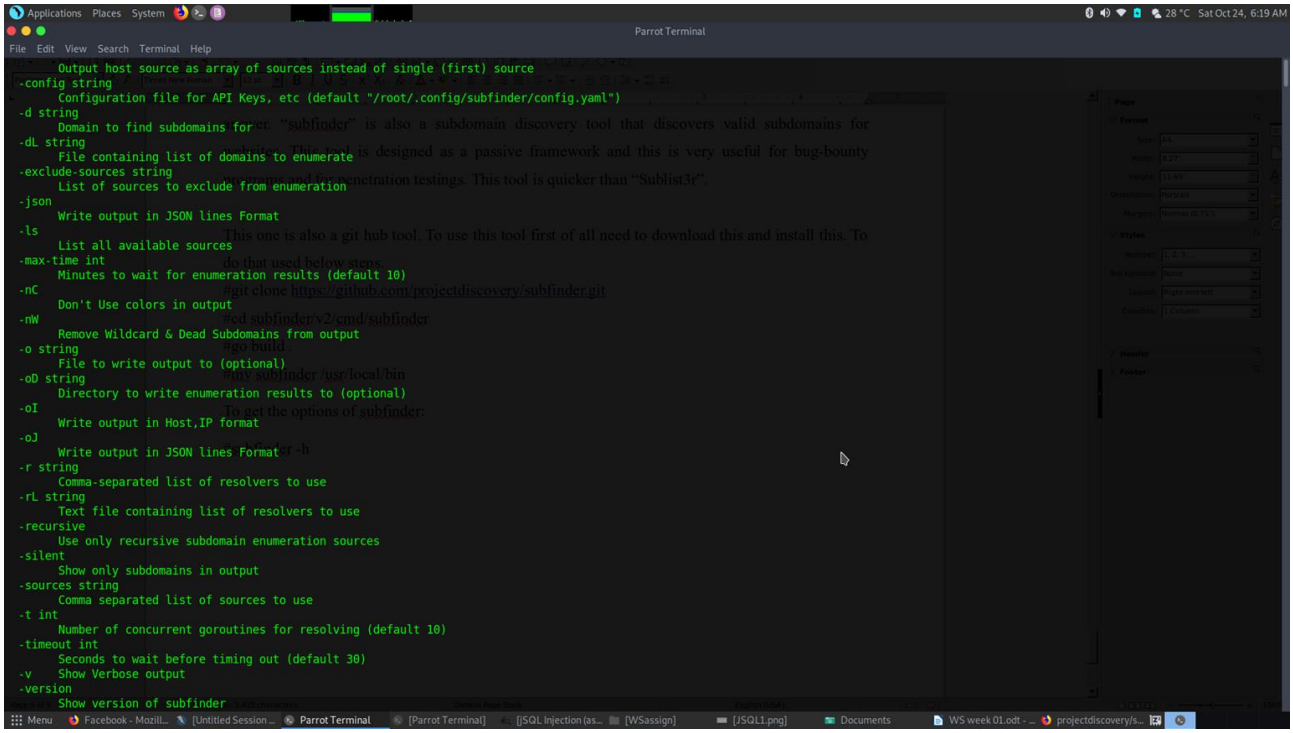
#go build .

#mv subfinder /usr/local/bin

To get the options of subfinder:

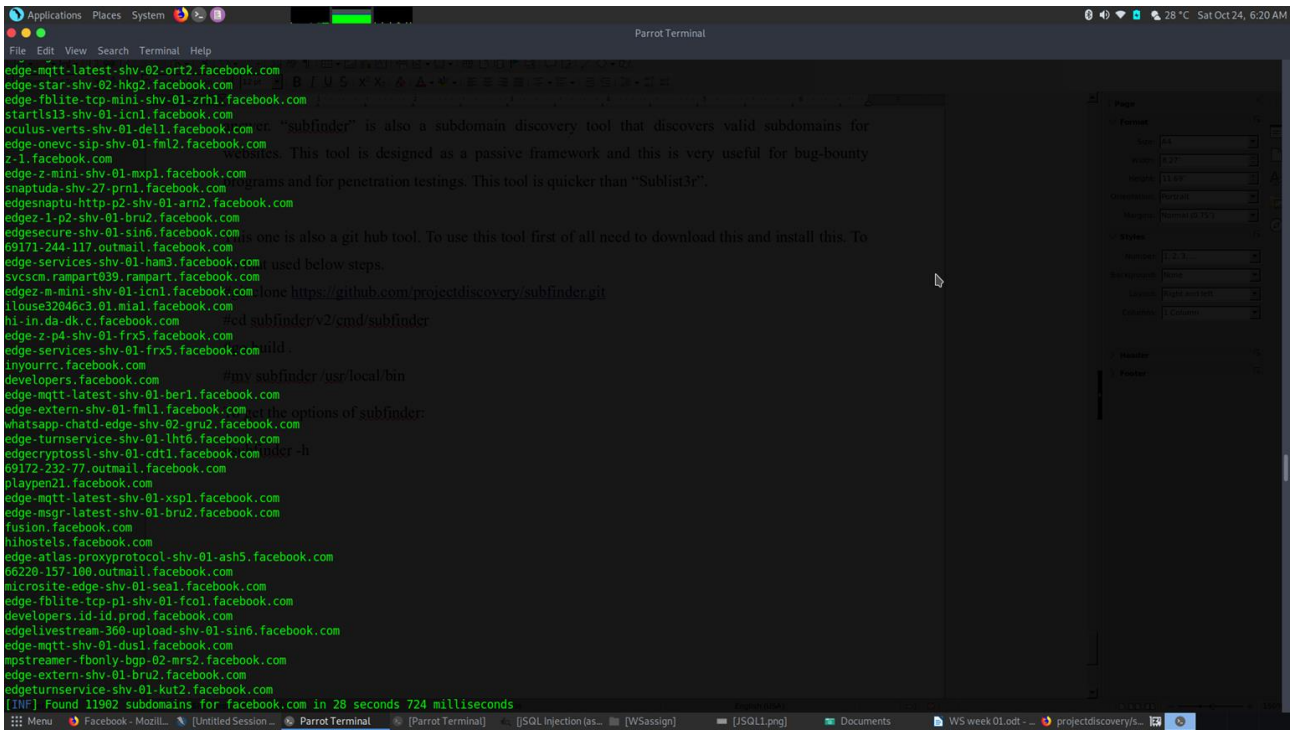
#subfinder -h





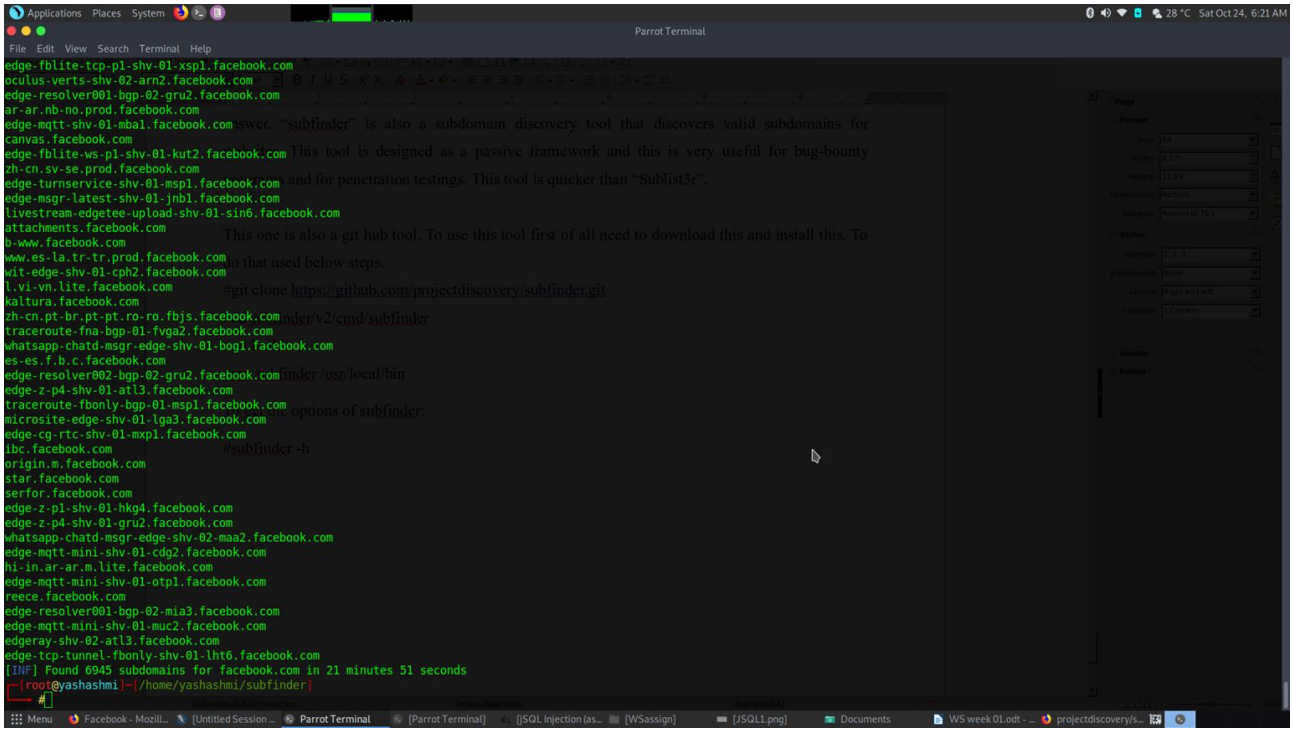
To find the number of subdomains:

#subfinder -d <url>



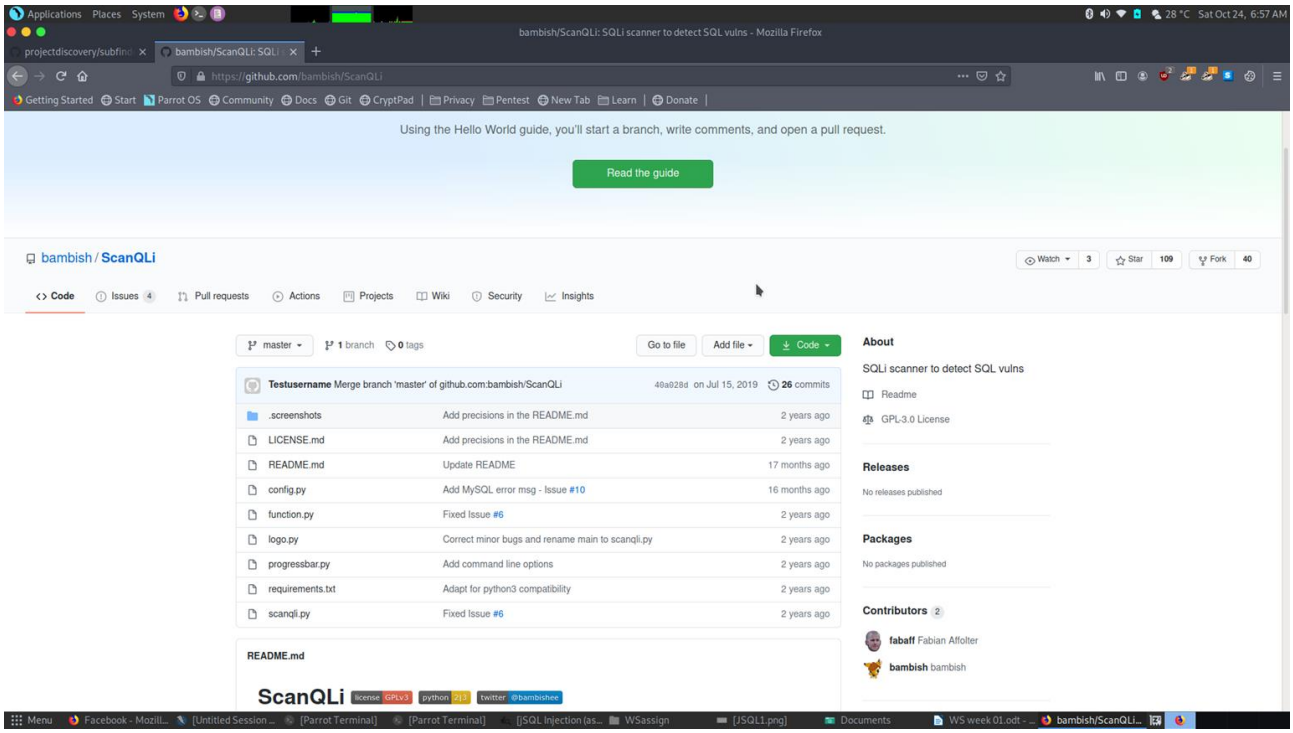
To find the number of subdomains except wildcard and dead subdomains:

#subfinder -d <url> -nW



scanQLI

After finding the number of subdomains, used scanQLI tool to scan the website for sql vulnerabilities. This tool was created using python language.



Downloaded this tool from github. To download and use:

#git clone https://github.com/bambish/ScanQLi

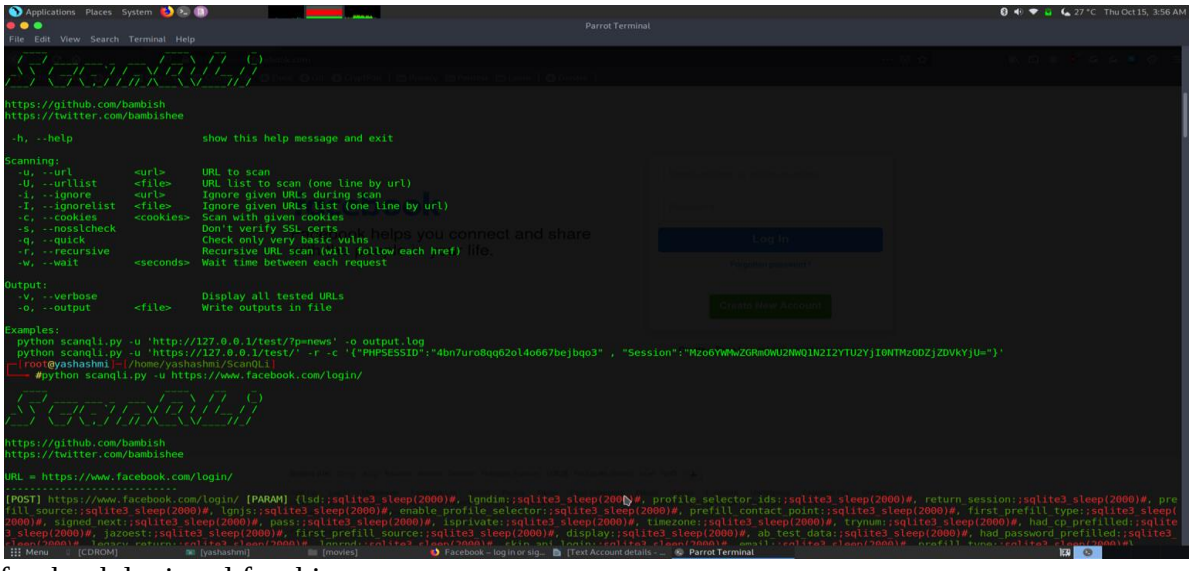
#apt install python-pip

#cd ScanQLi

#pip install -r requirements.txt

To use:

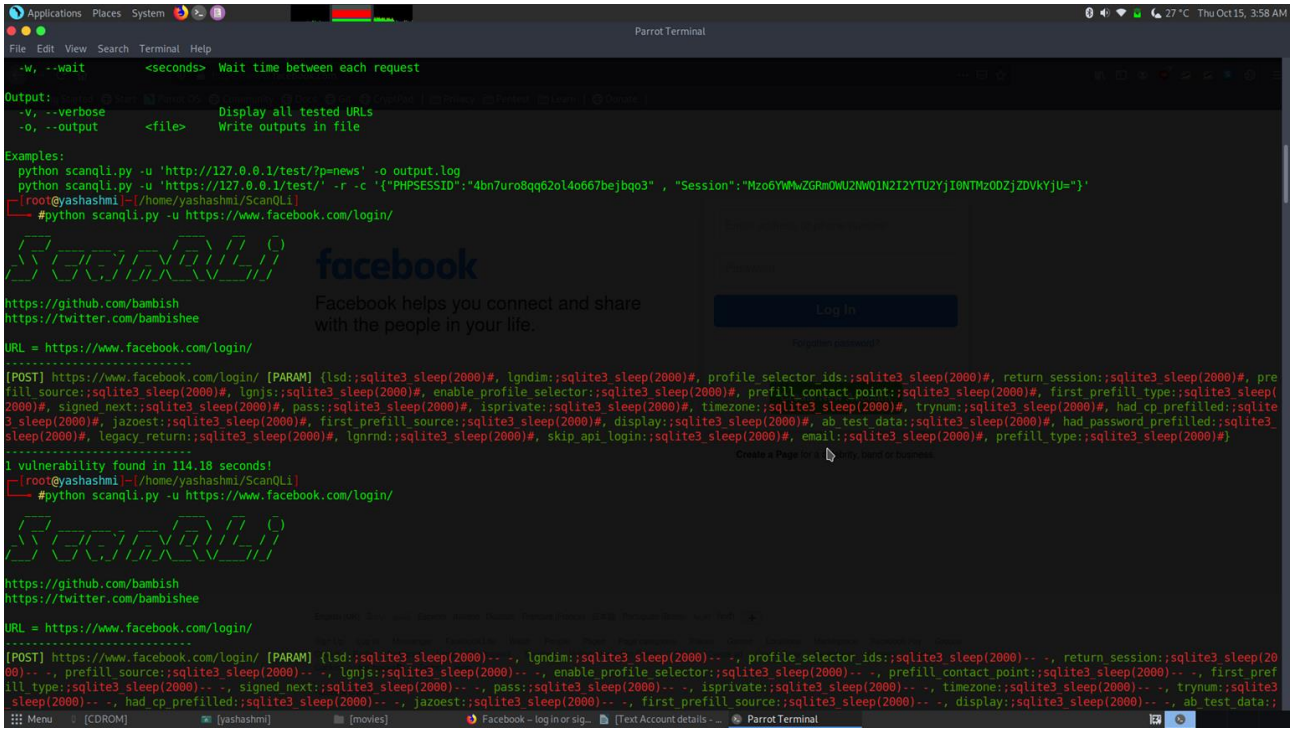
#python scanqli -u [URL] [OPTIONS]



Used the facebook login url for this scan.

#python scanQLi -u https://www.facebook.com/login/

By giving above command line, can scan the given url to check the url for sql vulnerabilities.

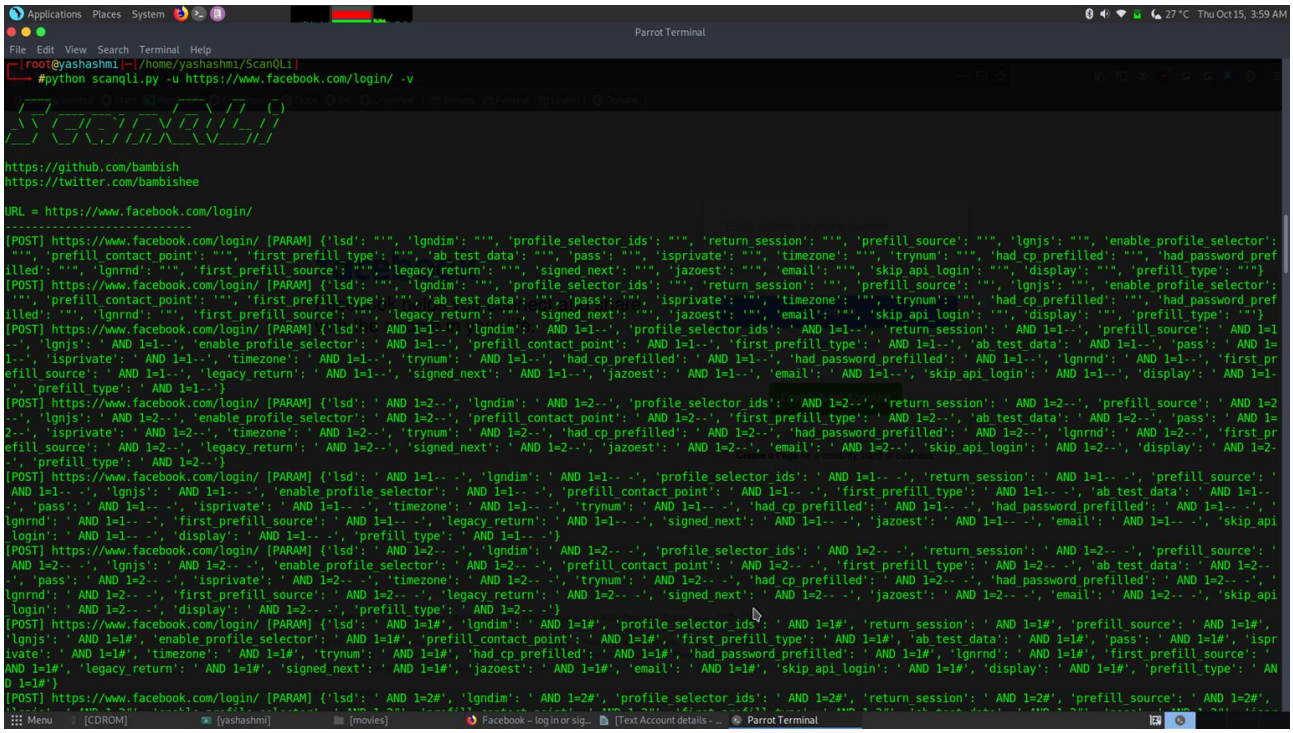


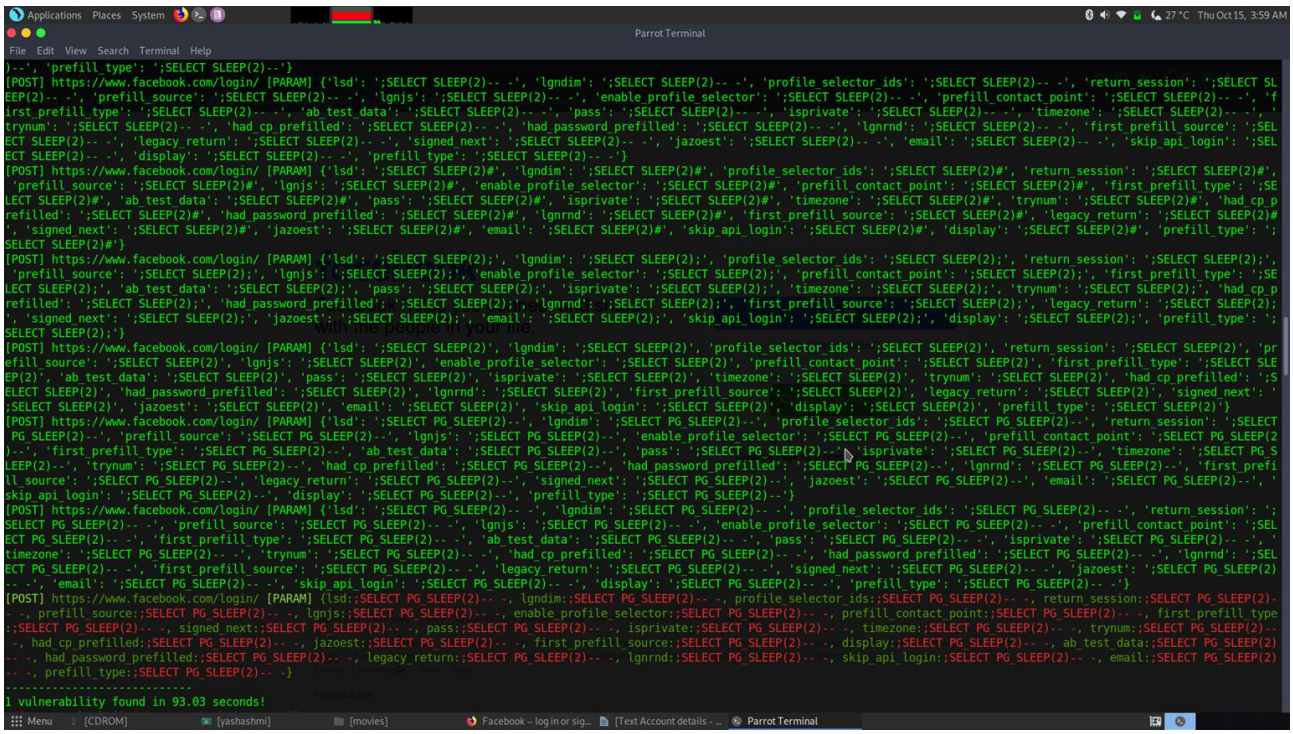
By using this tool Found one vulnerability.

After that changed the command:

#python scanQLi -u https://www.facebook.com/login/ -v

-v: It will display the all tested URLs.

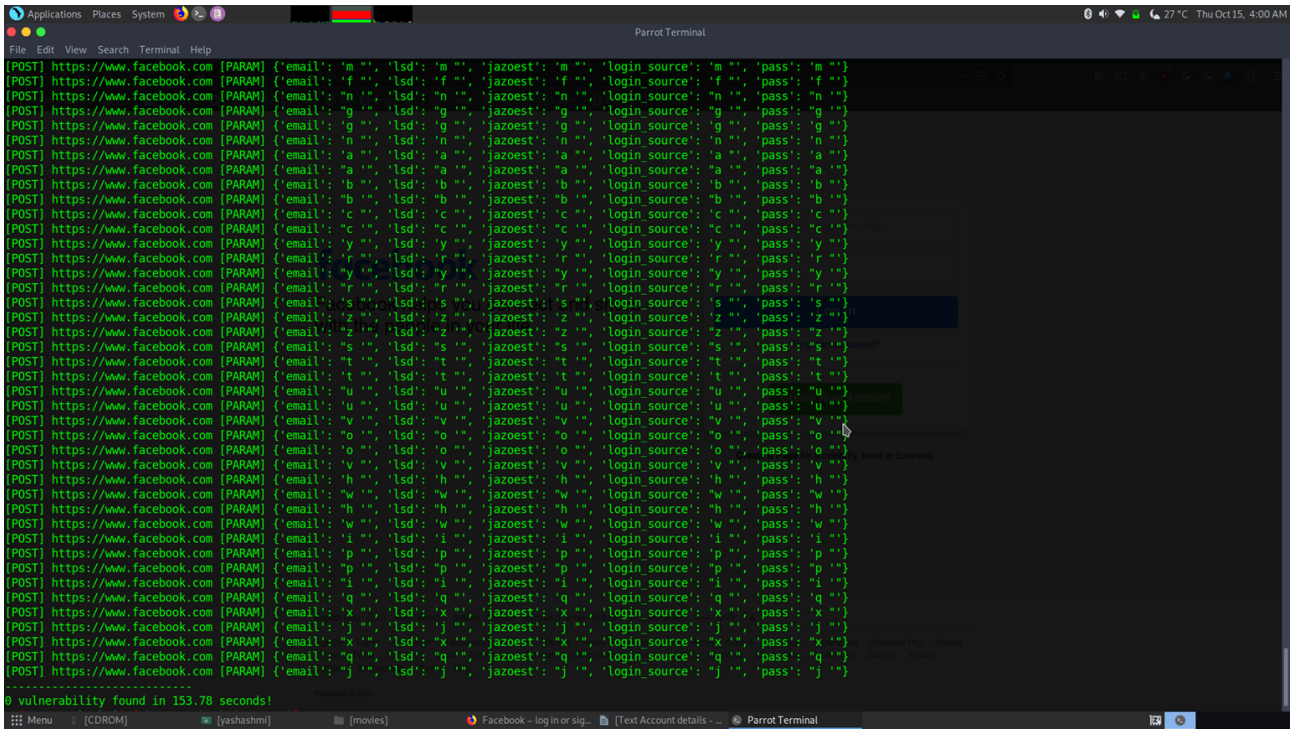




After that, changed the url:

#python scanQLi -u https://www.facebook.com

to check the scanner and the results.



It gave the answer as 0.

Using this tool, found one sql vulnerability in the facebook login URL.

nikto

The nikto web server scanner is a security application that can test thousands of potential security vulnerabilities on a web site. Dangerous files, misconfigured services, insecure scripts and other problems are included. It is open source and structured with plugins expanding the capabilities. These plugins are upgraded with new security tests regularly.

nikto isn't a stealthy weapon by any stretch. About 2000 HTTP GET requests would be made to the database server, generating a huge number of entries in the log files of the web servers. In fact, this noise is a perfect way to test an on-site intrusion detection system (IDS) that is in operation. A ikto scan should be detected by any web server log monitoring, host based intrusion detection (HIDS) or network-based intrusion detection (NIDS).

If several pages are hosted by the web server using virtual hosts. To get higher vulnerability coverage, can test each virtual host using nikto. In particular, checking the IP address and the host name of the site will be useful to ensure that all routes are checked for any insecure web applications and scripts. Because of the amount of security tests that this method does, depending on the speed of your web server, a search will take 45 minutes or even longer.

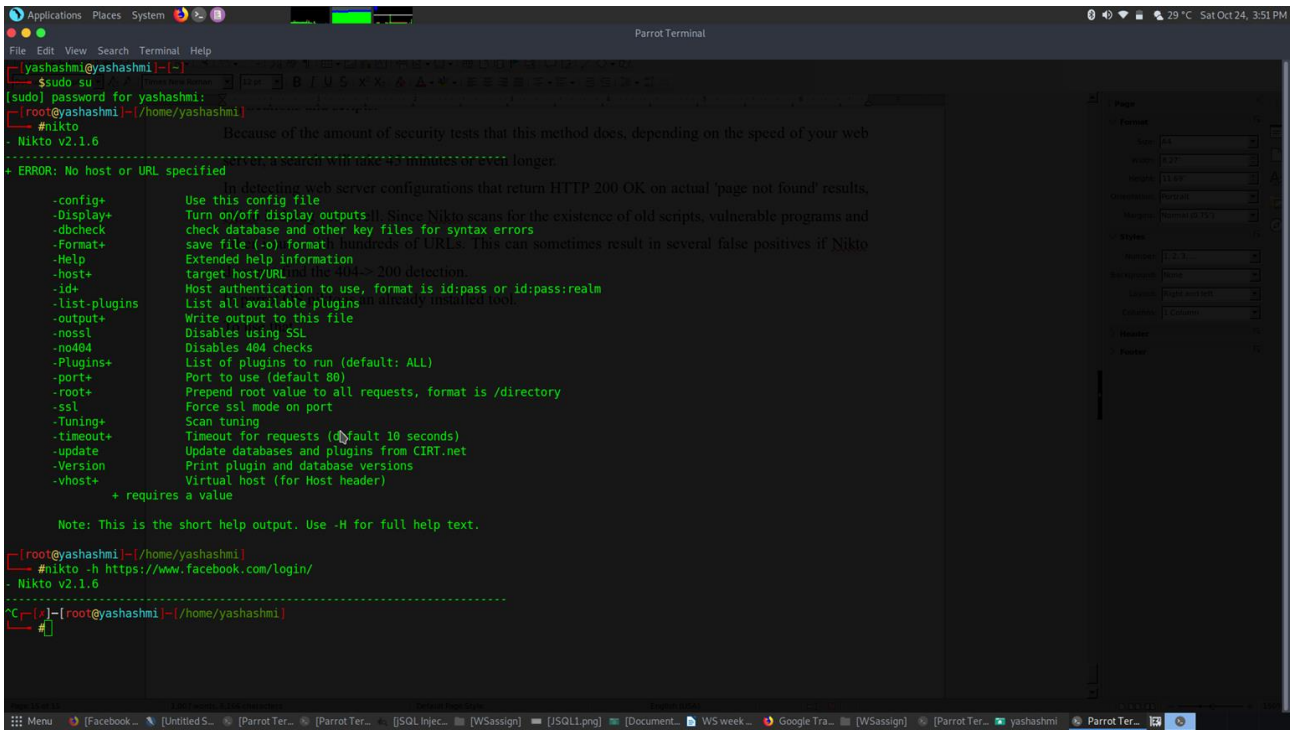
In detecting web server configurations that return HTTP 200 OK on actual 'page not found' results, nikto is doing very well. Since nikto scans for the existence of old scripts, vulnerable programs and other issues with hundreds of URLs. This can sometimes result in several false positives if nikto does not find the 404-> 200 detection.

In parrot OS nikto is an already installed tool.

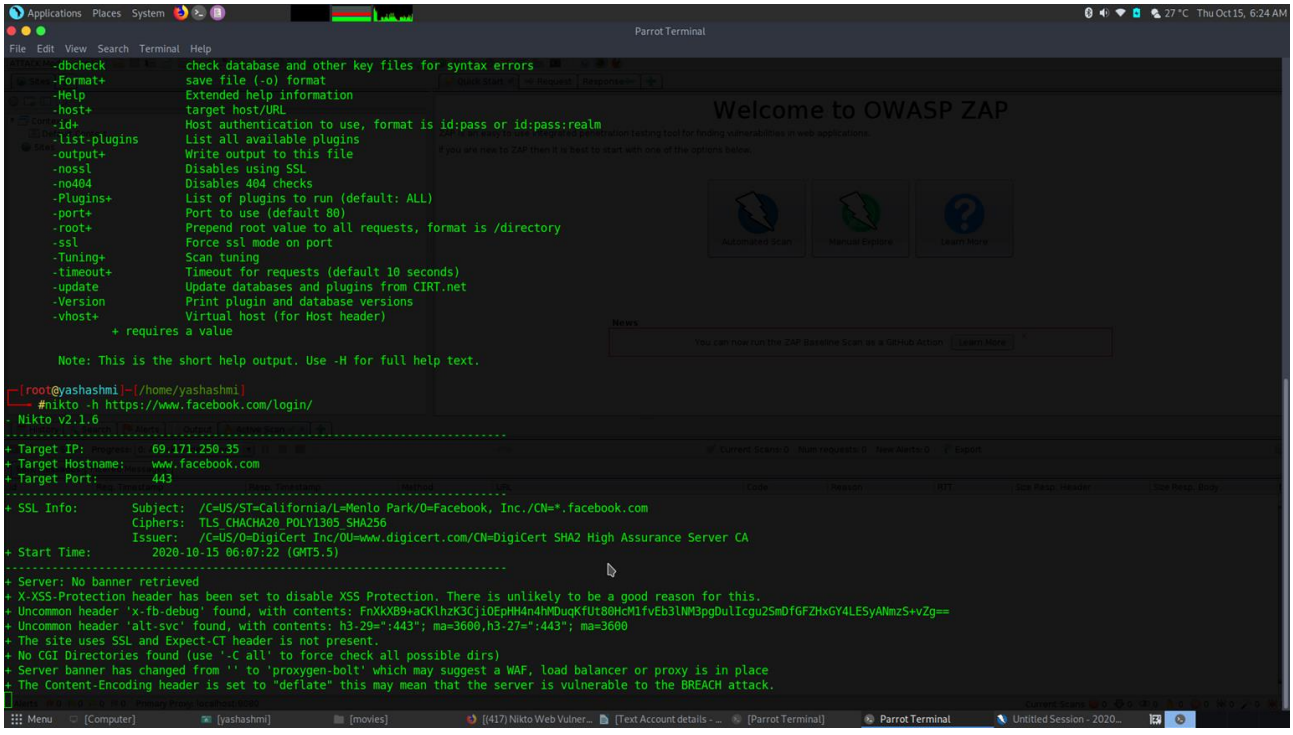
To use that,

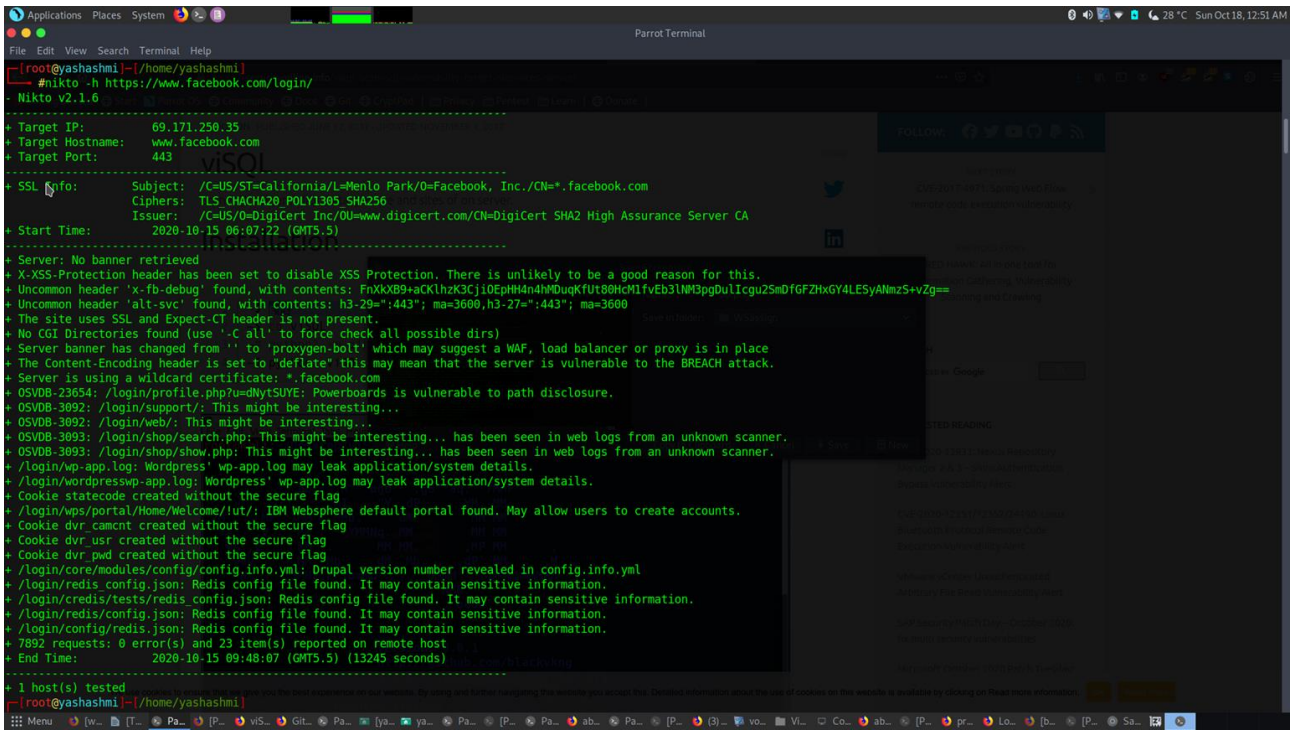
#nikto

Just give the above command



By using this tool, scanned the facebook login url.

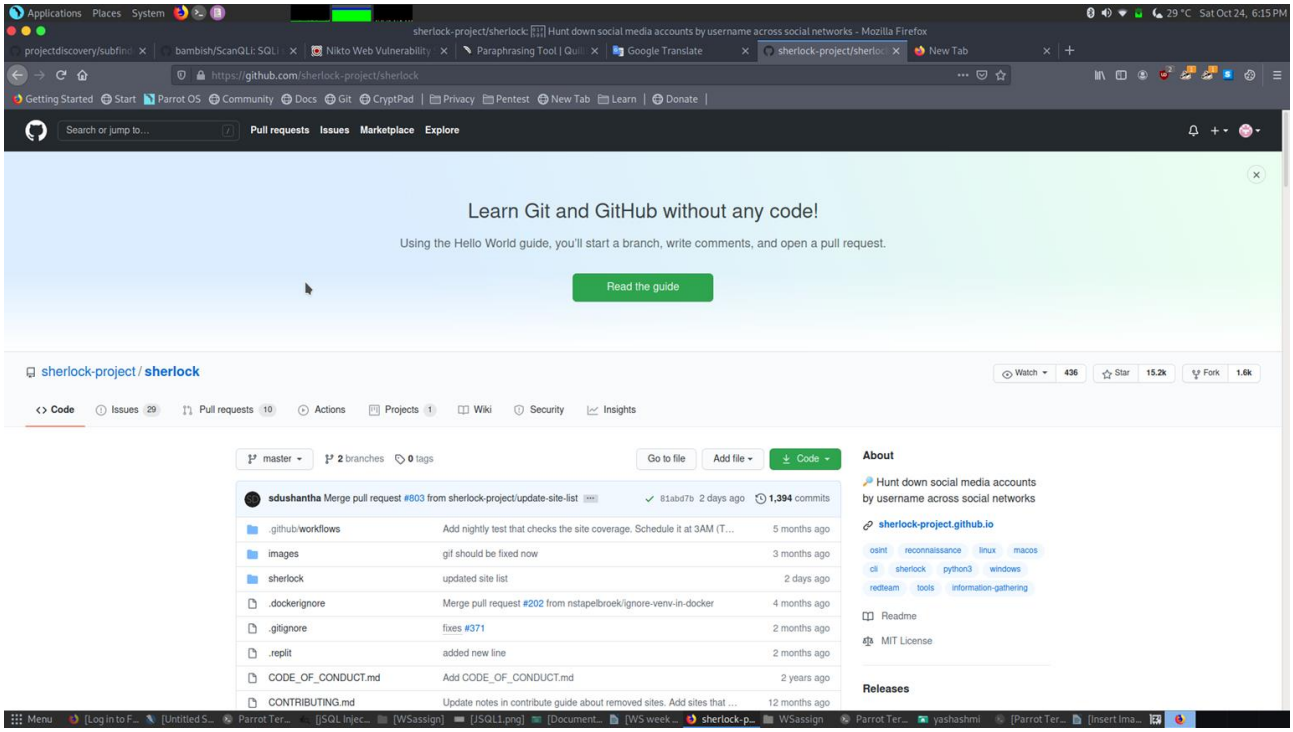




This gave several vulnerabilities with the given link.

Sherlock

Using this tool, can hunt down social media networks using username. Important thing of this tool is without having a facebook account, can find another one's face book account and view that account.



This tool, can download from github.

Method to install:

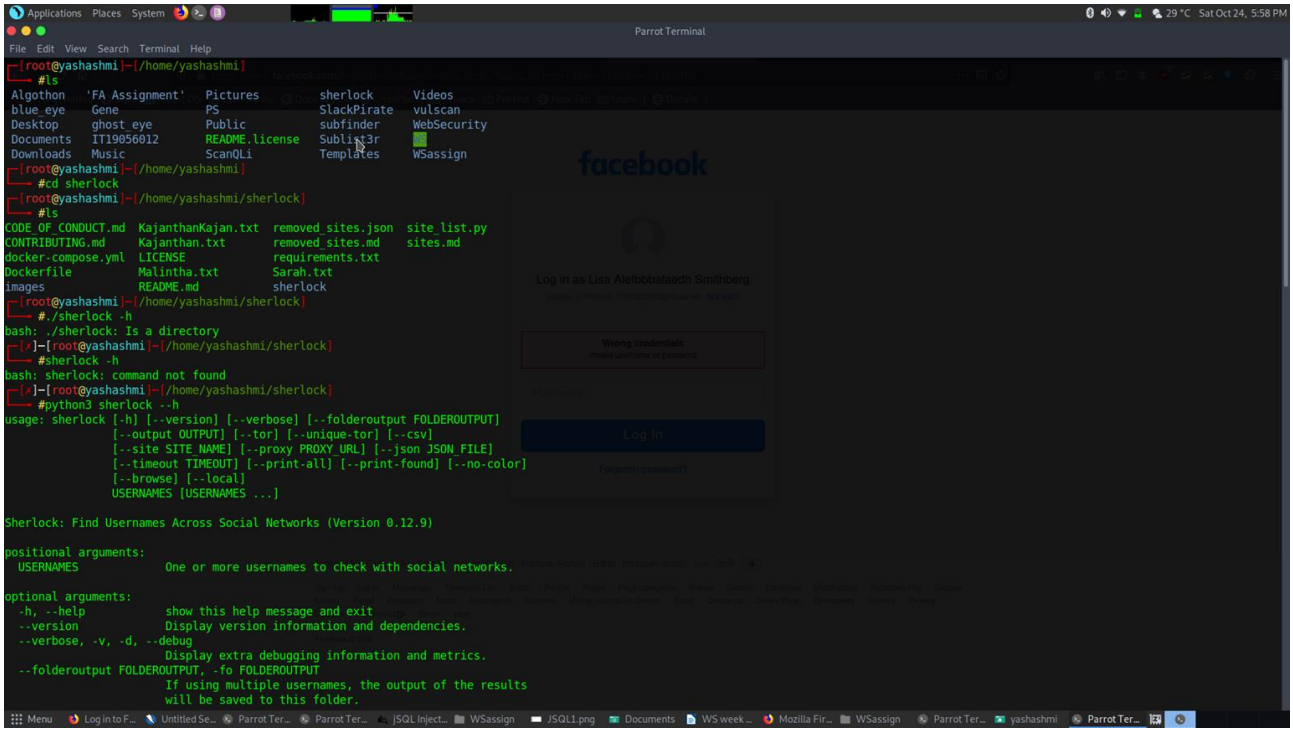
#git clone https://github.com/sherlock-project/sherlock.git

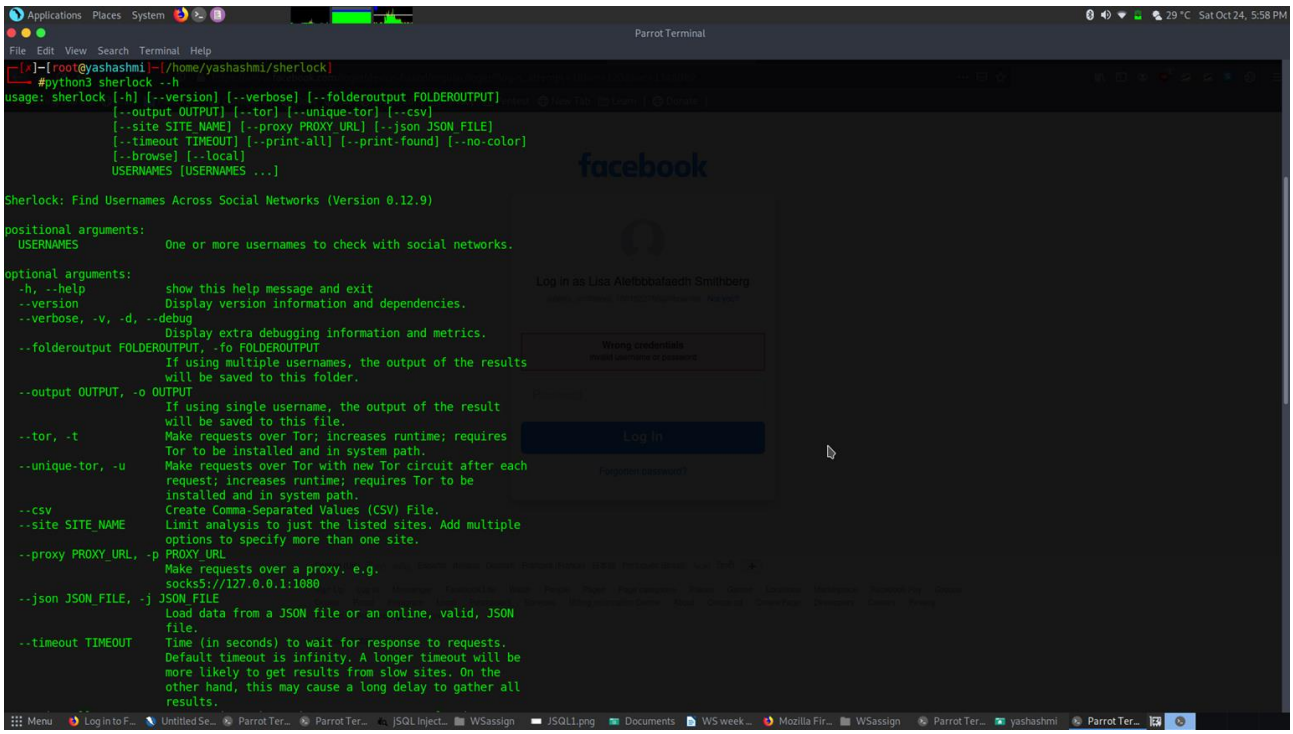
#cd sherlock

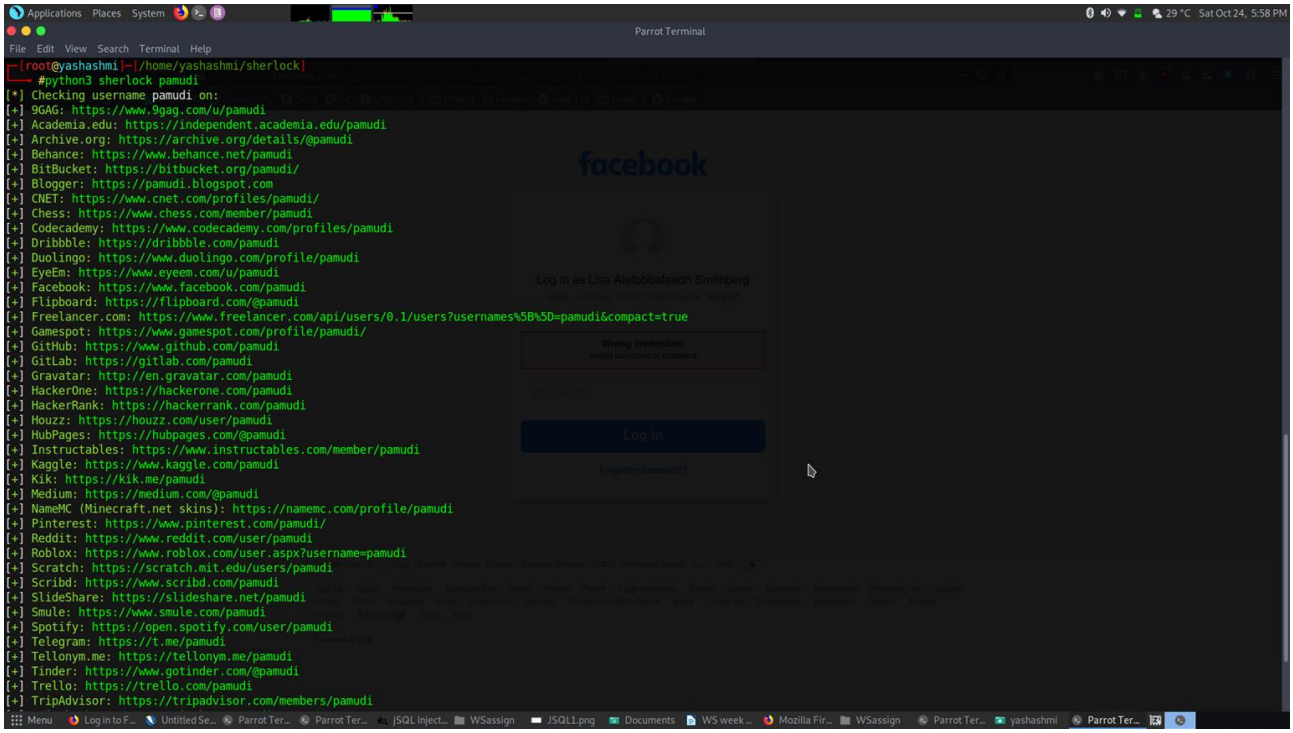
#python3 -m pip install -r requirements.txt

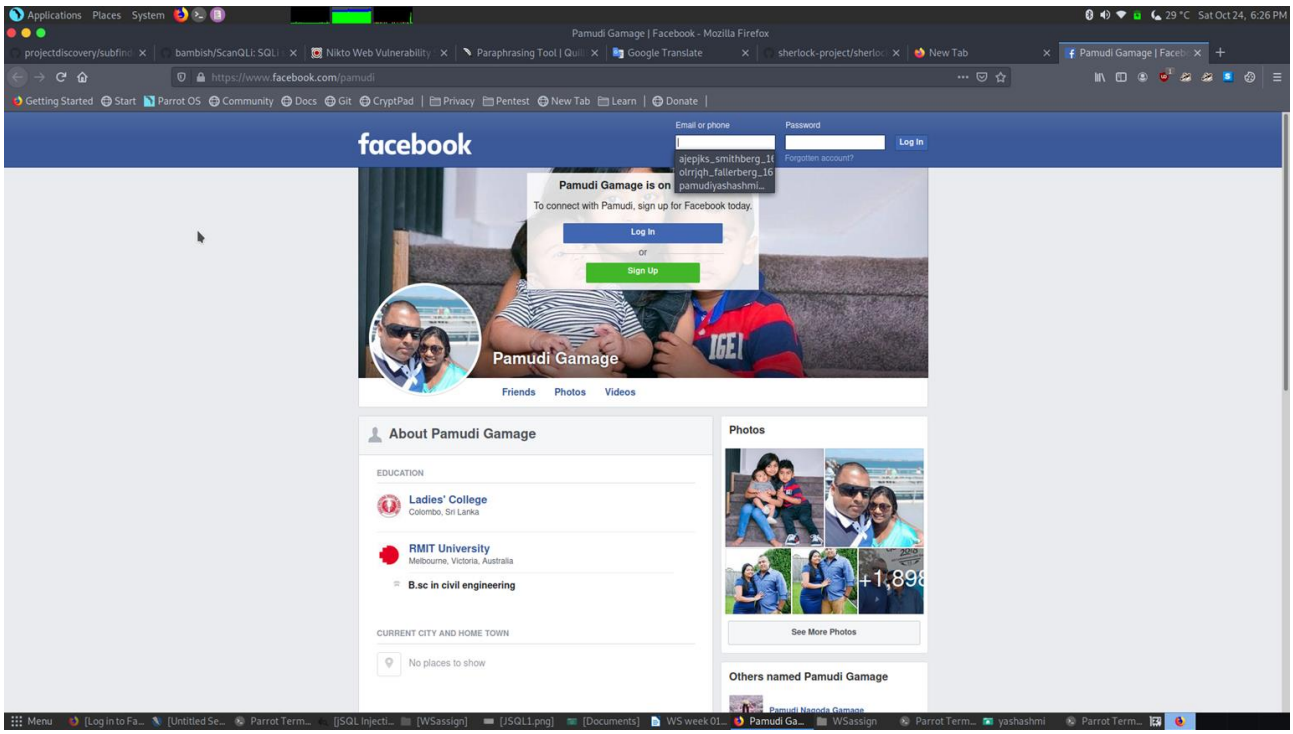
To use:

#python3 sherlock [username]




Applications Places System
29 °C Sat Oct 24, 5:58 PM
Parrot Terminal
File Edit View Search Terminal Help
┌[✗]-[root@yashashmi]-[/home/yashashmi/sherlock]
└─→ #python3 sherlock --h
usage: sherlock [-h] [--version] [--verbose] [--folderoutput FOLDEROUTPUT]
[--output OUTPUT] [--tor] [--unique-tor] [--csv]
[--site SITE_NAME] [--proxy PROXY_URL] [--json JSON_FILE]
[--timeout TIMEOUT] [--print-all] [--print-found] [--no-color]
[--browse] [--local]
USERNAMES [USERNAMES ...]
Sherlock: Find Usernames Across Social Networks (Version 0.12.9)
positional arguments:
USERNAMES One or more usernames to check with social networks.
optional arguments:
-h, --help show this help message and exit
--version Display version information and dependencies.
--verbose, -v, -d, --debug
Display extra debugging information and metrics.
--folderoutput FOLDEROUTPUT, -fo FOLDEROUTPUT
If using multiple usernames, the output of the results
will be saved to this folder.
--output OUTPUT, -o OUTPUT
If using single username, the output of the result
will be saved to this file.
--tor, -t Make requests over Tor; increases runtime; requires
Tor to be installed and in system path.
--unique-tor, -u Make requests over Tor with new Tor circuit after each
request; increases runtime; requires Tor to be
installed and in system path.
--csv Create Comma-Separated Values (CSV) File.
--site SITE_NAME Limit analysis to just the listed sites. Add multiple
options to specify more than one site.
--proxy PROXY_URL, -p PROXY_URL
Make requests over a proxy. e.g.
socks5://127.0.0.1:1080
--json JSON_FILE, -j JSON_FILE
Load data from a JSON file or an online, valid, JSON
file.
--timeout TIMEOUT Time (in seconds) to wait for response to requests.
Default timeout is infinity. A longer timeout will be
more likely to get results from slow sites. On the
other hand, this may cause a long delay to gather all
results.
Menu

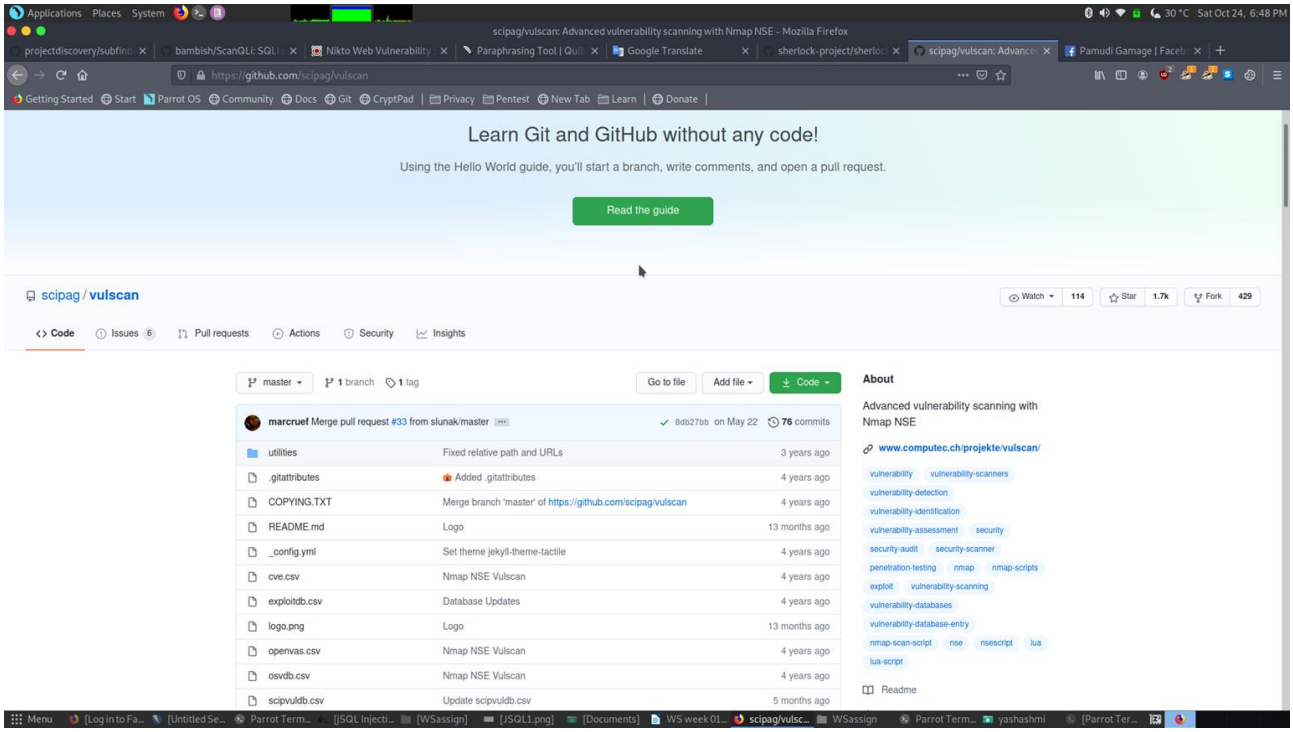

Applications Places System
29 °C Sat Oct 24, 5:58 PM
Parrot Terminal
File Edit View Search Terminal Help
┌[root@yashashmi]-[/home/yashashmi/sherlock]
└─→ #python3 sherlock pamudi
[*] Checking username pamudi on:
[+] 9GAG: https://www.9gag.com/u/pamudi
[+] Academia.edu: https://independent.academia.edu/pamudi
[+] Archive.org: https://archive.org/details/@pamudi
[+] Behance: https://www.behance.net/pamudi
[+] BitBucket: https://bitbucket.org/pamudi/
[+] Blogger: https://pamudi.blogspot.com
[+] CNET: https://www.cnet.com/profiles/pamudi/
[+] Chess: https://www.chess.com/member/pamudi
[+] Codecademy: https://www.codecademy.com/profiles/pamudi
[+] Dribbble: https://dribbble.com/pamudi
[+] Duolingo: https://www.duolingo.com/profile/pamudi
[+] EyeEm: https://www.eyeem.com/u/pamudi
[+] Facebook: https://www.facebook.com/pamudi
[+] Flipboard: https://flipboard.com/@pamudi
[+] Freelancer.com: https://www.freelancer.com/api/users/0.1/users?usernames%5B%5D=pamudi&compact=true
[+] Gamespot: https://www.gamespot.com/profile/pamudi/
[+] GitHub: https://www.github.com/pamudi
[+] GitLab: https://gitlab.com/pamudi
[+] Gravatar: http://en.gravatar.com/pamudi
[+] HackerOne: https://hackerone.com/pamudi
[+] HackerRank: https://hackerrank.com/pamudi
[+] Houzz: https://houzz.com/user/pamudi
[+] HubPages: https://hubpages.com/@pamudi
[+] Instructables: https://www.instructables.com/member/pamudi
[+] Kaggle: https://www.kaggle.com/pamudi
[+] Kik: https://kik.me/pamudi
[+] Medium: https://medium.com/@pamudi
[+] NameMC (Minecraft.net skins): https://namemc.com/profile/pamudi
[+] Pinterest: https://www.pinterest.com/pamudi/
[+] Reddit: https://www.reddit.com/user/pamudi
[+] Roblox: https://www.roblox.com/user.aspx?username=pamudi
[+] Scratch: https://scratch.mit.edu/users/pamudi
[+] Scribd: https://www.scribd.com/pamudi
[+] SlideShare: https://slideshare.net/pamudi
[+] Smule: https://www.smule.com/pamudi
[+] Spotify: https://open.spotify.com/user/pamudi
[+] Telegram: https://t.me/pamudi
[+] Tellonym.me: https://tellonym.me/pamudi
[+] Tinder: https://www.gotinder.com/@pamudi
[+] Trello: https://trello.com/pamudi
[+] TripAdvisor: https://tripadvisor.com/members/pamudi
Menu



By studying above screen shots, can see that sherlock tool can hunt a facebook account using a username and without having a facebook account also can view that hunted account.

Vulscan

Vulscan is a module that improves a vulnerability scanner with nmap. The nmap alternative -sV allows identification of versions per service, which is used to determine possible defects according to the product found. In an offline version of VulDB, the data is looked up.



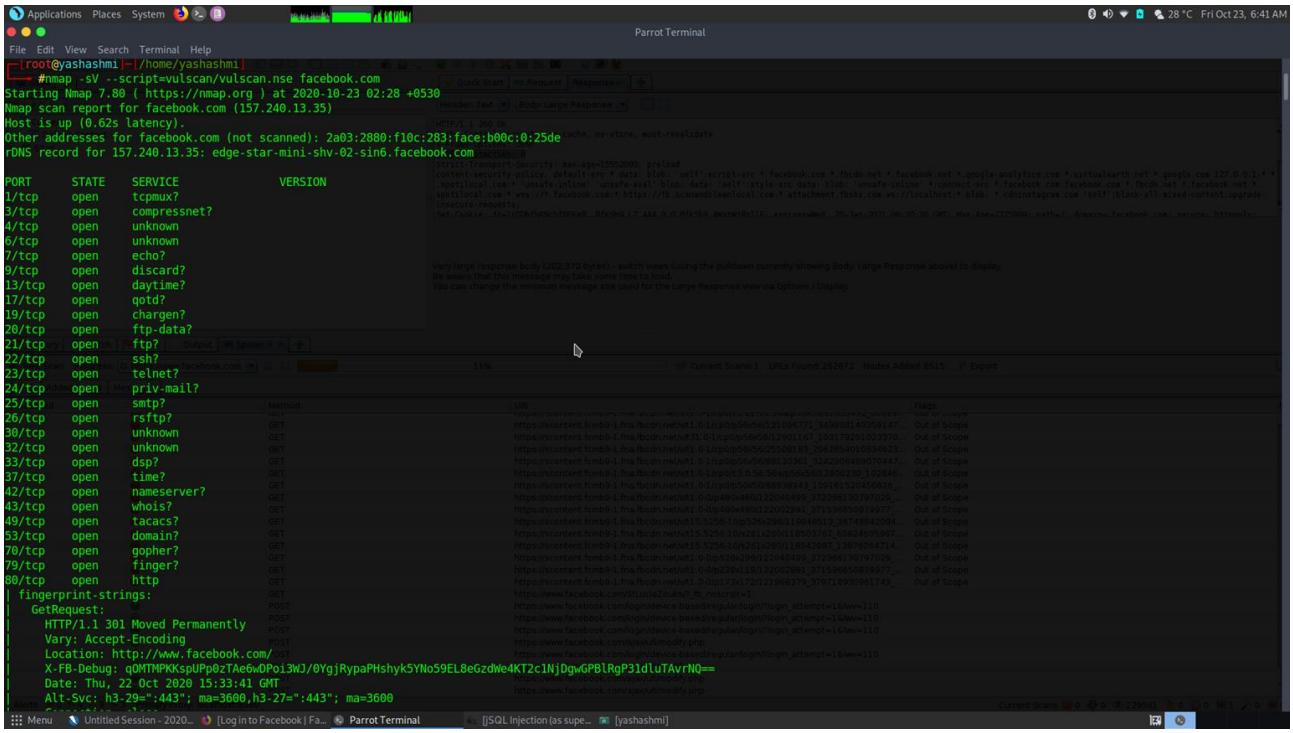
To install:

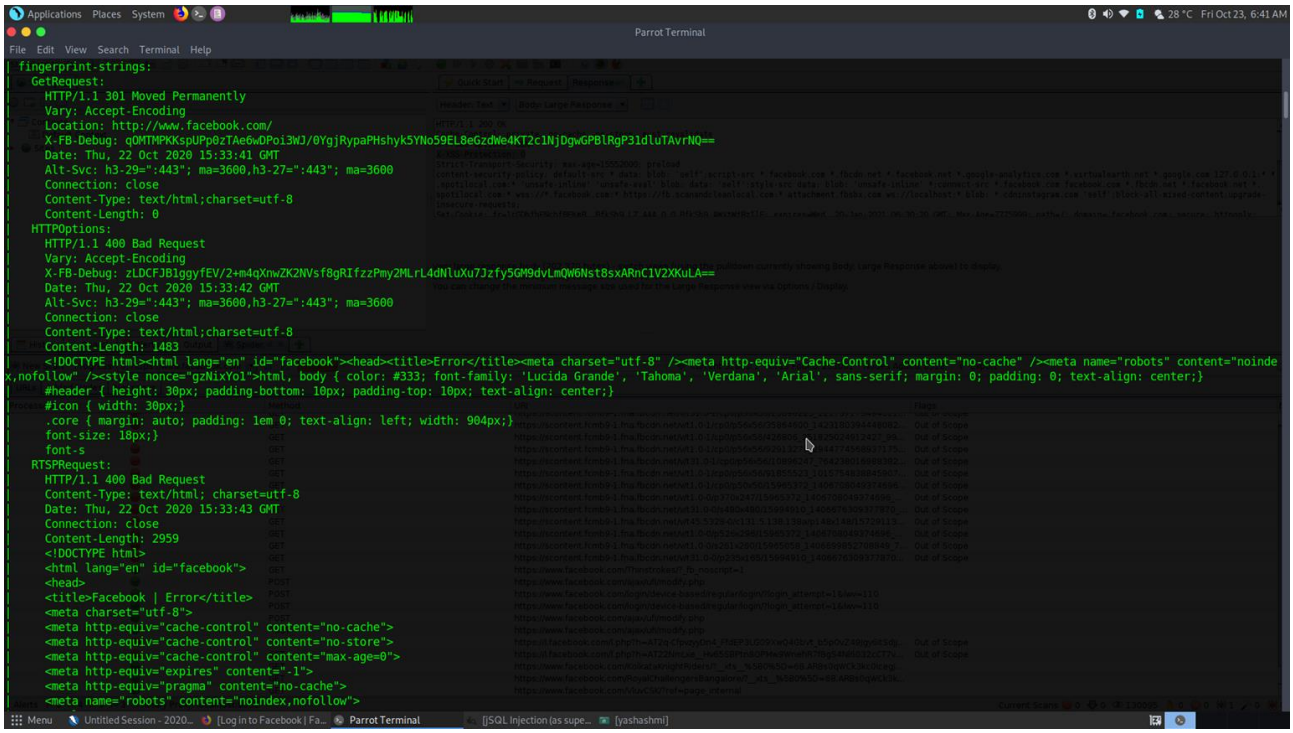
#git clone https://github.com/scipag/vulscan.git

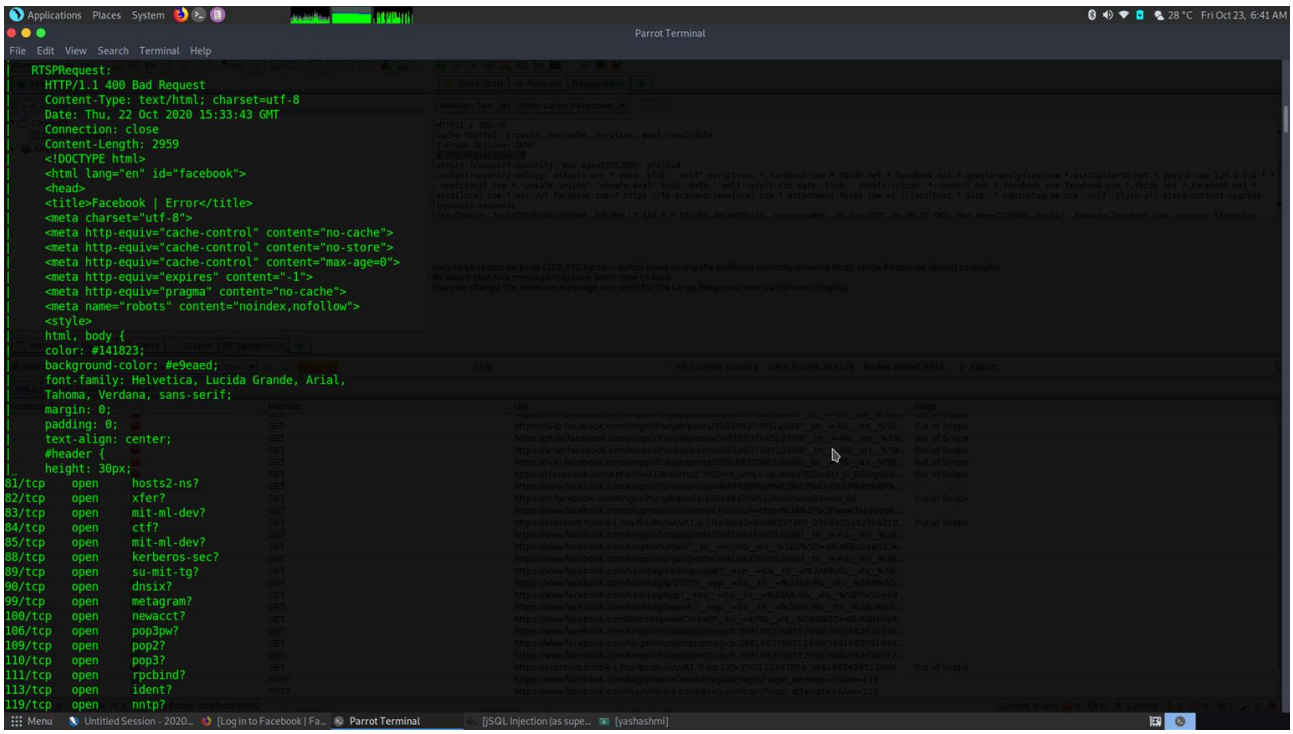
#ln -s `pwd`/scipag_vulscan /usr/share/nmap/scripts/vulscan

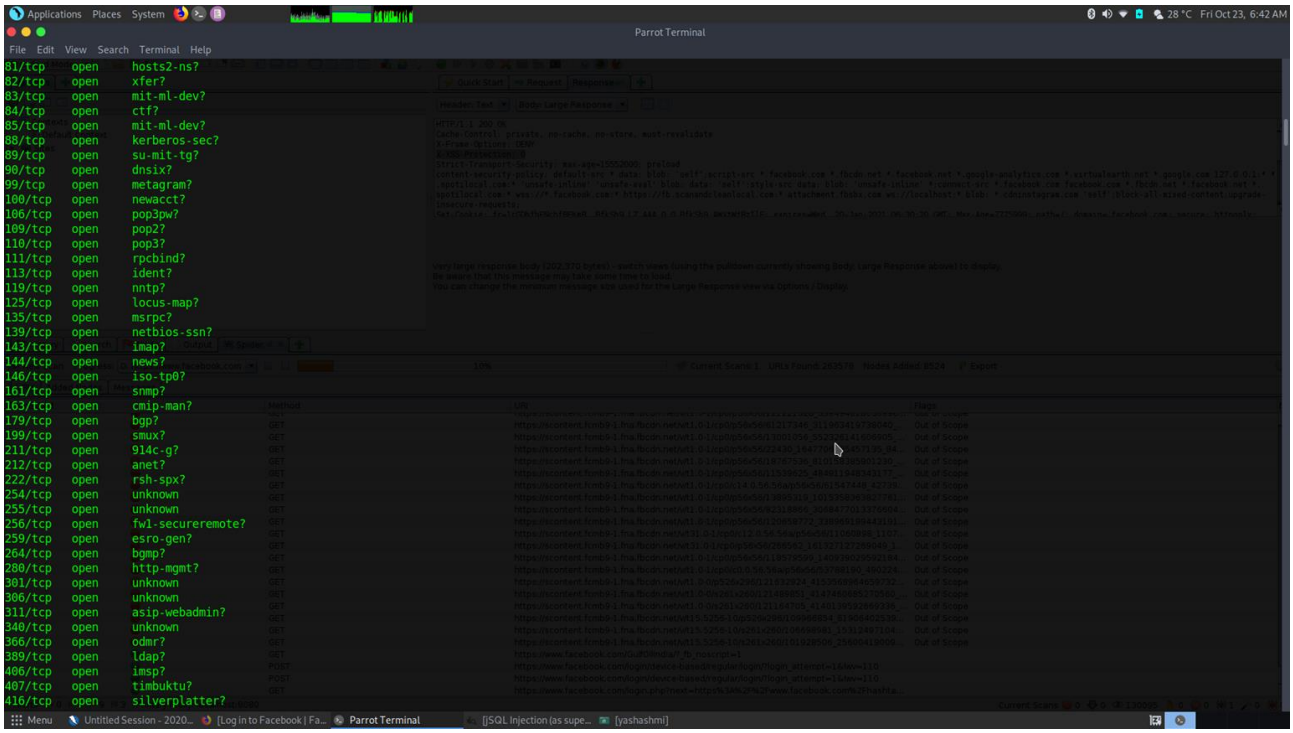
To use:

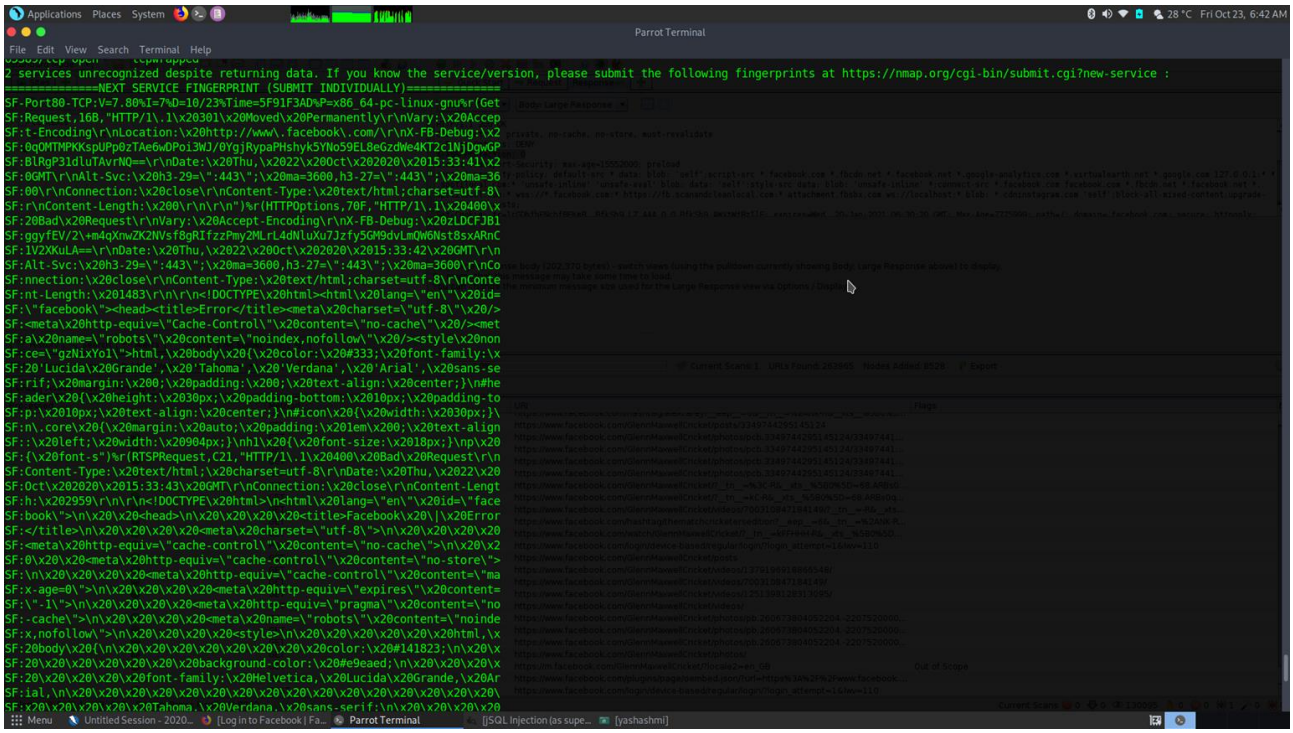
#nmap -sV --script=vulscan/vulscan.nse www.facebook.com

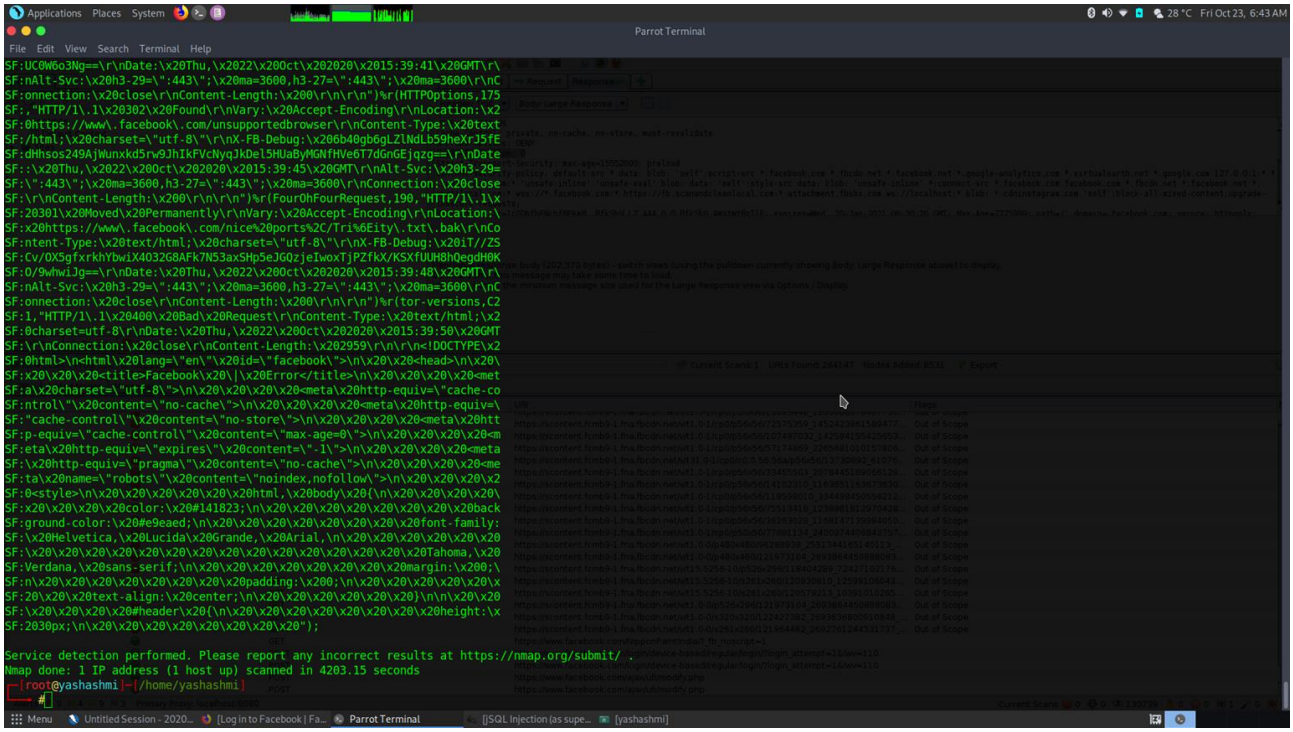





Applications Places System
28 °C Fri Oct 23, 6:41 AM
Parrot Terminal
File Edit View Search Terminal Help
RTSPRequest:
HTTP/1.1 400 Bad Request
Content-Type: text/html; charset=utf-8
Date: Thu, 22 Oct 2020 15:33:43 GMT
Connection: close
Content-Length: 2959
<!DOCTYPE html>
<html lang="en" id="facebook">
<head>
<title>Facebook | Error</title>
<meta charset="utf-8">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="cache-control" content="no-store">
<meta http-equiv="cache-control" content="max-age=0">
<meta http-equiv="expires" content="-1">
<meta http-equiv="pragma" content="no-cache">
<meta name="robots" content="noindex,nofollow">
<style>
html, body {
color: #141823;
background-color: #e9eaed;
font-family: Helvetica, Lucida Grande, Arial,
Tahoma, Verdana, sans-serif;
margin: 0;
padding: 0;
text-align: center;
#header {
height: 30px;
81/tcp open hosts2-ns?
82/tcp open xfer?
83/tcp open mit-ml-dev?
84/tcp open ctf?
85/tcp open mit-ml-dev?
88/tcp open kerberos-sec?
89/tcp open su-mit-tg?
90/tcp open dnsix?
99/tcp open metagram?
100/tcp open newacct?
106/tcp open pop3pw?
109/tcp open pop2?
110/tcp open pop3?
111/tcp open rpcbind?
113/tcp open ident?
119/tcp open nntp?
Menu
Untitled Session - 2020...
[Log in to Facebook | Fa...
Parrot Terminal
[jSQL Injection (as supe...
[yashashmi]


Applications Places System
28 °C Fri Oct 23, 6:42 AM
Parrot Terminal
File Edit View Search Terminal Help
81/tcp open hosts2-ns?
82/tcp open xfer?
83/tcp open mit-ml-dev?
84/tcp open ctf?
85/tcp open mit-ml-dev?
88/tcp open kerberos-sec?
89/tcp open su-mit-tg?
90/tcp open dnsix?
99/tcp open metagram?
100/tcp open newacct?
106/tcp open pop3pw?
109/tcp open pop2?
110/tcp open pop3?
111/tcp open rpcbind?
113/tcp open ident?
119/tcp open nntp?
125/tcp open locus-map?
135/tcp open msrpc?
139/tcp open netbios-ssn?
143/tcp open imap?
144/tcp open news?
146/tcp open iso-tp0?
161/tcp open snmp?
163/tcp open cmip-man?
179/tcp open bgp?
199/tcp open smux?
211/tcp open 914c-g?
212/tcp open anet?
222/tcp open rsh-spx?
254/tcp open unknown
255/tcp open unknown
256/tcp open fw1-secureremote?
259/tcp open esro-gen?
264/tcp open bgmp?
280/tcp open http-mgmt?
301/tcp open unknown
306/tcp open unknown
311/tcp open asip-webadmin?
340/tcp open unknown
366/tcp open odmr?
389/tcp open ldap?
406/tcp open imsp?
407/tcp open timbuktu?
416/tcp open silverplatter?
Menu
Untitled Session - 2020...
[Log in to Facebook | Fa...
Parrot Terminal
[jSQL Injection (as supe...
[yashashmi]


Applications Places System
28 °C Fri Oct 23, 6:42 AM
Parrot Terminal
File Edit View Search Terminal Help
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port80-TCP:V=7.80%I=7%D=10/23%Time=5F91F3AD%P=x86_64-pc-linux-gnu%r(Get
SF:Request,16B,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\nVary:\x20Accep
SF:t-Encoding\r\nLocation:\x20http://www\.facebook\.com/\r\nX-FB-Debug:\x2
SF:0qOMTMPKKspUPp0zTAe6wOPoi3WJ/0YgjRypaPHshyk5YNo59EL8eGzdWe4KT2c1NjDgwGP
SF:BlRgP31dluTAvrNQ==\r\nDate:\x20Thu,\x2022\x20Oct\x202020\x2015:33:41\x2
SF:0GMT\r\nAlt-Svc:\x20h3-29=\":443\";\x20ma=3600,h3-27=\":443\";\x20ma=36
SF:00\r\nConnection:\x20close\r\nContent-Type:\x20text/html;charset=utf-8\
SF:r\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,70F,"HTTP/1\.1\x20400\x
SF:20Bad\x20Request\r\nVary:\x20Accept-Encoding\r\nX-FB-Debug:\x20zLDCFJB1
SF:ggyfEV/2\+m4qXnwZK2NVsf8gRIfzzPmy2MLrL4dNluXu7Jzfy5GM9dvLmQW6Nst8sxARnC
SF:1V2XKuLA==\r\nDate:\x20Thu,\x2022\x20Oct\x202020\x2015:33:42\x20GMT\r\n
SF:Alt-Svc:\x20h3-29=\":443\";\x20ma=3600,h3-27=\":443\";\x20ma=3600\r\nCo
SF:nnection:\x20close\r\nContent-Type:\x20text/html;charset=utf-8\r\nConte
SF:nt-Length:\x201483\r\n\r\n<!DOCTYPE\x20html><html\x20lang=\"en\"\x20id=
SF:\"facebook\"><head><title>Error</title><meta\x20charset=\"utf-8\"\x20/>
SF:<meta\x20http-equiv=\"Cache-Control\"\x20content=\"no-cache\"\x20/><met
SF:a\x20name=\"robots\"\x20content=\"noindex,nofollow\"\x20/><style\x20non
SF:ce=\"gzNixYo1\">html,\x20body\x20{\x20color:\x20#333;\x20font-family:\x
SF:20'Lucida\x20Grande',\x20'Tahoma',\x20'Verdana',\x20'Arial',\x20sans-se
SF:rif;\x20margin:\x200;\x20padding:\x200;\x20text-align:\x20center;}\n#he
SF:ader\x20{\x20height:\x2030px;\x20padding-bottom:\x2010px;\x20padding-to
SF:p:\x2010px;\x20text-align:\x20center;}\n#icon\x20{\x20width:\x2030px;}\
SF:n\.core\x20{\x20margin:\x20auto;\x20padding:\x201em\x200;\x20text-align
SF::\x20left;\x20width:\x20904px;}\nh1\x20{\x20font-size:\x2018px;}\np\x20
SF:{\x20font-s")%r(RTSPRequest,C21,"HTTP/1\.1\x20400\x20Bad\x20Request\r\n
SF:Content-Type:\x20text/html;\x20charset=utf-8\r\nDate:\x20Thu,\x2022\x20
SF:Oct\x202020\x2015:33:43\x20GMT\r\nConnection:\x20close\r\nContent-Lengt
SF:h:\x202959\r\n\r\n<!DOCTYPE\x20html>\n<html\x20lang=\"en\"\x20id=\"face
SF:book\">\n\x20\x20<head>\n\x20\x20\x20\x20<title>Facebook\x20\|\x20Error
SF:</title>\n\x20\x20\x20\x20<meta\x20charset=\"utf-8\">\n\x20\x20\x20\x20
SF:<meta\x20http-equiv=\"cache-control\"\x20content=\"no-cache\">\n\x20\x2
SF:0\x20\x20<meta\x20http-equiv=\"cache-control\"\x20content=\"no-store\">
SF:\n\x20\x20\x20\x20<meta\x20http-equiv=\"cache-control\"\x20content=\"ma
SF:x-age=0\">\n\x20\x20\x20\x20<meta\x20http-equiv=\"expires\"\x20content=
SF:\"-1\">\n\x20\x20\x20\x20<meta\x20http-equiv=\"pragma\"\x20content=\"no
SF:-cache\">\n\x20\x20\x20\x20<meta\x20name=\"robots\"\x20content=\"noinde
SF:x,nofollow\">\n\x20\x20\x20\x20<style>\n\x20\x20\x20\x20\x20\x20html,\x
SF:20body\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20color:\x20#141823;\n\x20\x
SF:20\x20\x20\x20\x20\x20\x20background-color:\x20#e9eaed;\n\x20\x20\x20\x
SF:20\x20\x20\x20\x20font-family:\x20Helvetica,\x20Lucida\x20Grande,\x20Ar
SF:ial,\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20Tahoma,\x20Verdana,\x20sans-serif;\n\x20\x20\x20\x20
Menu
Untitled Session - 2020...
[Log in to Facebook | Fa...
Parrot Terminal
[jSQL Injection (as supe...
[yashashmi]


Applications Places System
28 °C Fri Oct 23, 6:43 AM
Parrot Terminal
File Edit View Search Terminal Help
SF:UC0W6o3Ng==\r\nDate:\x20Thu,\x2022\x20Oct\x202020\x2015:39:41\x20GMT\r\
SF:nAlt-Svc:\x20h3-29=\":443\";\x20ma=3600,h3-27=\":443\";\x20ma=3600\r\nC
SF:onnection:\x20close\r\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,175
SF:,"HTTP/1\.1\x20302\x20Found\r\nVary:\x20Accept-Encoding\r\nLocation:\x2
SF:0https://www\.facebook\.com/unsupportedbrowser\r\nContent-Type:\x20text
SF:/html;\x20charset=\"utf-8\"\r\nX-FB-Debug:\x206b40gb6gLZlNdLb59heXrJ5fE
SF:dHhsos249AjWunxkd5rw9JhIkFVcNyqJkDel5HUaByMGNfHVe6T7dGnGEjqzg==\r\nDate
SF::\x20Thu,\x2022\x20Oct\x202020\x2015:39:45\x20GMT\r\nAlt-Svc:\x20h3-29=
SF:\":443\";\x20ma=3600,h3-27=\":443\";\x20ma=3600\r\nConnection:\x20close
SF:\r\nContent-Length:\x200\r\n\r\n")%r(FourOhFourRequest,190,"HTTP/1\.1\x
SF:20301\x20Moved\x20Permanently\r\nVary:\x20Accept-Encoding\r\nLocation:\
SF:x20https://www\.facebook\.com/nice%20ports%2C/Tri%6Eity\.txt\.bak\r\nCo
SF:ntent-Type:\x20text/html;\x20charset=\"utf-8\"\r\nX-FB-Debug:\x20iT//ZS
SF:Cv/OX5gfxrkhYbwiX4O32G8AFk7N53axSHp5eJGQzjeIwoxTjPZfkX/KSXfUUH8hQegdH0K
SF:O/9whwiJg==\r\nDate:\x20Thu,\x2022\x20Oct\x202020\x2015:39:48\x20GMT\r\
SF:nAlt-Svc:\x20h3-29=\":443\";\x20ma=3600,h3-27=\":443\";\x20ma=3600\r\nC
SF:onnection:\x20close\r\nContent-Length:\x200\r\n\r\n")%r(tor-versions,C2
SF:1,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/html;\x2
SF:0charset=utf-8\r\nDate:\x20Thu,\x2022\x20Oct\x202020\x2015:39:50\x20GMT
SF:\r\nConnection:\x20close\r\nContent-Length:\x202959\r\n\r\n<!DOCTYPE\x2
SF:0html>\n<html\x20lang=\"en\"\x20id=\"facebook\">\n\x20\x20<head>\n\x20\
SF:x20\x20\x20<title>Facebook\x20\|\x20Error</title>\n\x20\x20\x20\x20<met
SF:a\x20charset=\"utf-8\">\n\x20\x20\x20\x20<meta\x20http-equiv=\"cache-co
SF:ntrol\"\x20content=\"no-cache\">\n\x20\x20\x20\x20<meta\x20http-equiv=\
SF:"cache-control\"\x20content=\"no-store\">\n\x20\x20\x20\x20<meta\x20htt
SF:p-equiv=\"cache-control\"\x20content=\"max-age=0\">\n\x20\x20\x20\x20<m
SF:eta\x20http-equiv=\"expires\"\x20content=\"-1\">\n\x20\x20\x20\x20<meta
SF:\x20http-equiv=\"pragma\"\x20content=\"no-cache\">\n\x20\x20\x20\x20<me
SF:ta\x20name=\"robots\"\x20content=\"noindex,nofollow\">\n\x20\x20\x20\x2
SF:0<style>\n\x20\x20\x20\x20\x20\x20html,\x20body\x20{\n\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20color:\x20#141823;\n\x20\x20\x20\x20\x20\x20\x20\x20back
SF:ground-color:\x20#e9eaed;\n\x20\x20\x20\x20\x20\x20\x20\x20font-family:
SF:\x20Helvetica,\x20Lucida\x20Grande,\x20Arial,\n\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20Tahoma,\x20
SF:Verdana,\x20sans-serif;\n\x20\x20\x20\x20\x20\x20\x20\x20margin:\x200;\
SF:n\x20\x20\x20\x20\x20\x20\x20\x20padding:\x200;\n\x20\x20\x20\x20\x20\x
SF:20\x20\x20text-align:\x20center;\n\x20\x20\x20\x20\x20\x20}\n\n\x20\x20
SF:\x20\x20\x20\x20#header\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20height:\x
SF:2030px;\n\x20\x20\x20\x20\x20\x20\x20\x20");

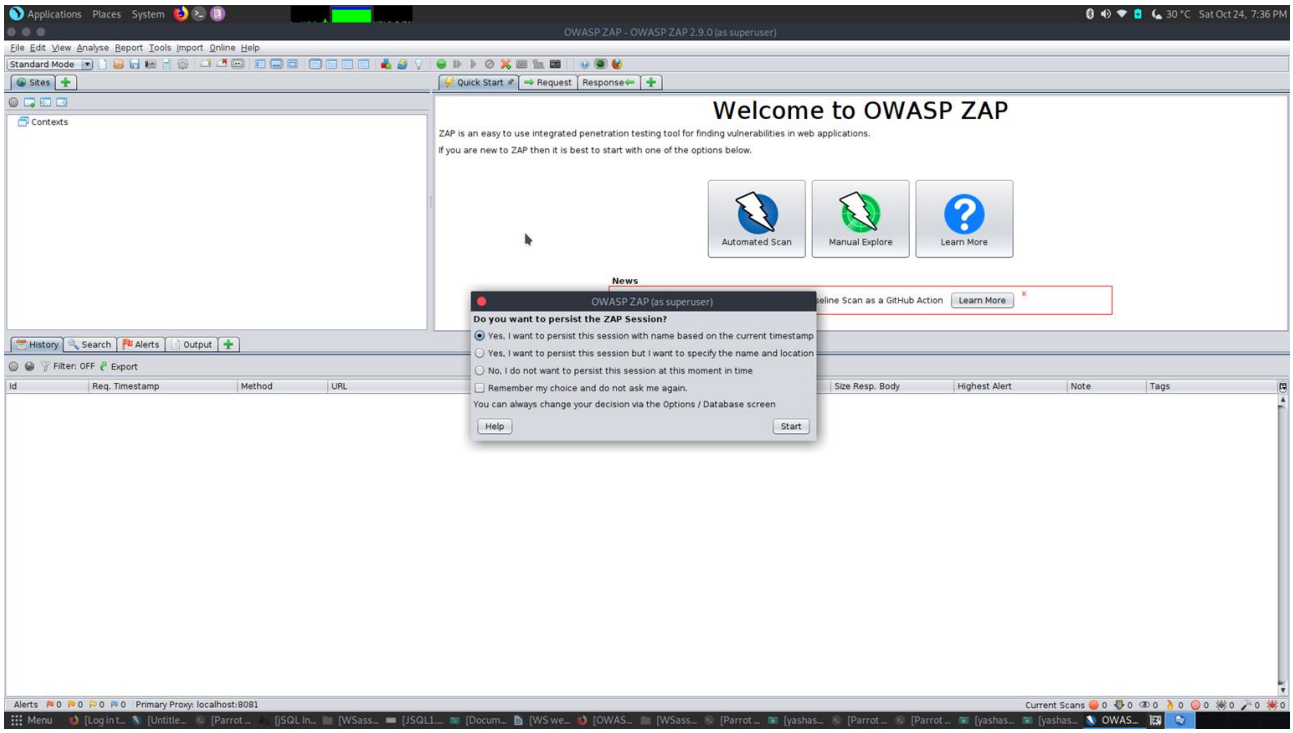
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4203.15 seconds
[root@yashashmi]~[/home/yashashmi]
#
Menu
Untitled Session - 2020...
[Log in to Facebook | Fa...
Parrot Terminal
[jSQL Injection (as supe...
[yashashmi]

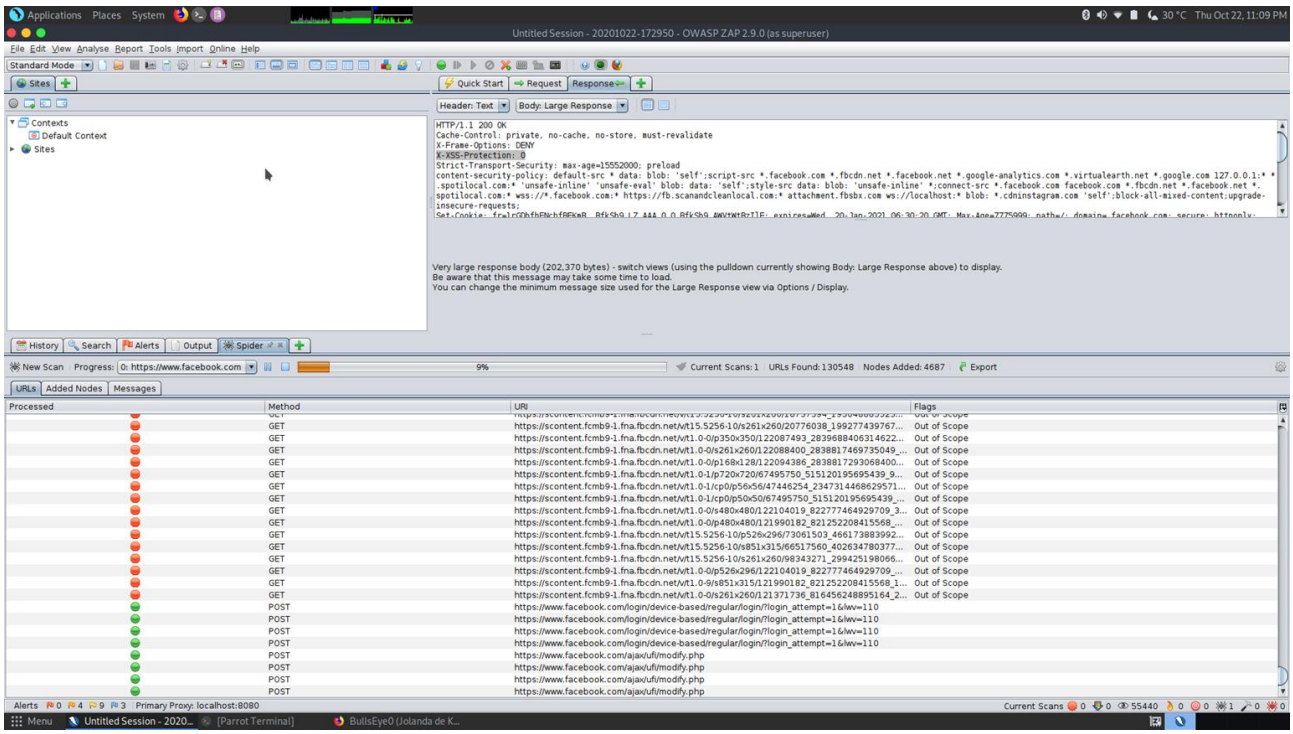
OWASP ZAP

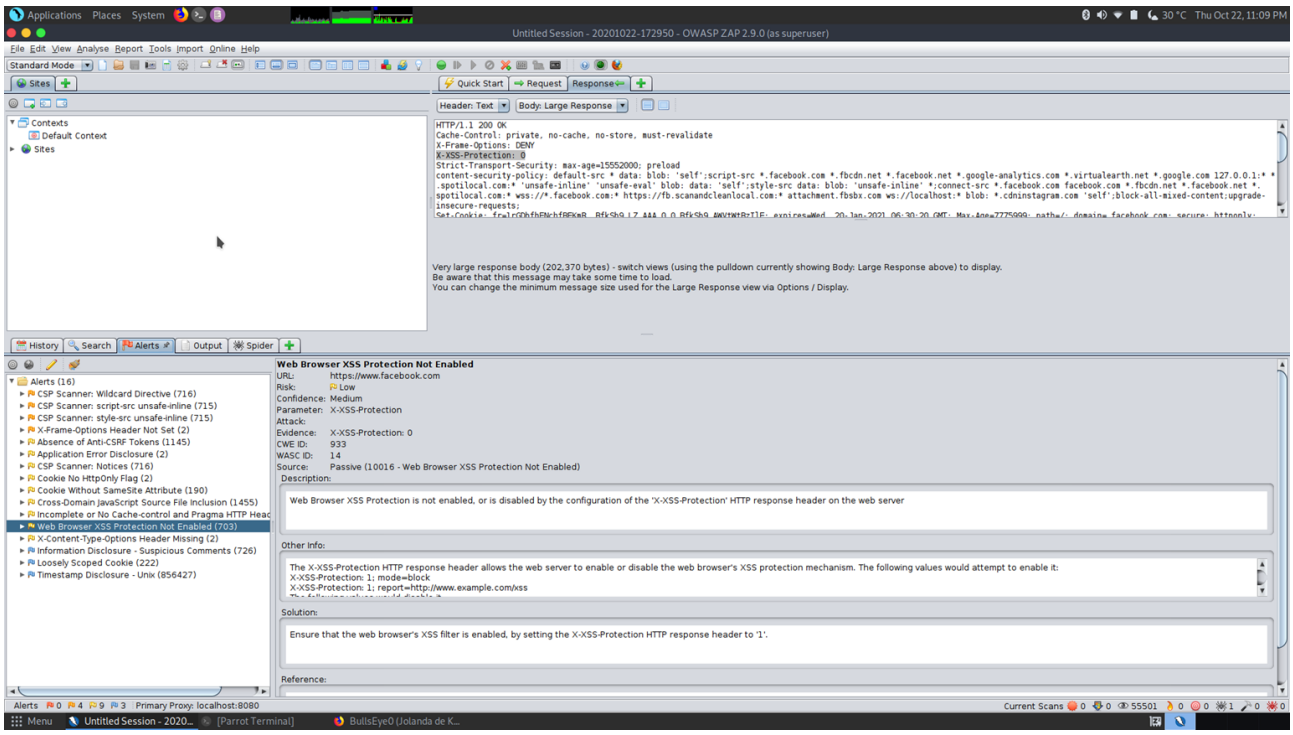
One of the most common web application security testing tools in the world is the OWASP Zed Attack Proxy (ZAP). As an open source project, it is made available for free, and is contributed to and maintained by OWASP. The Open Web Application Security Project (OWASP) is a non-profit, vendor-neutral organization of volunteers committed to creating more stable web applications.

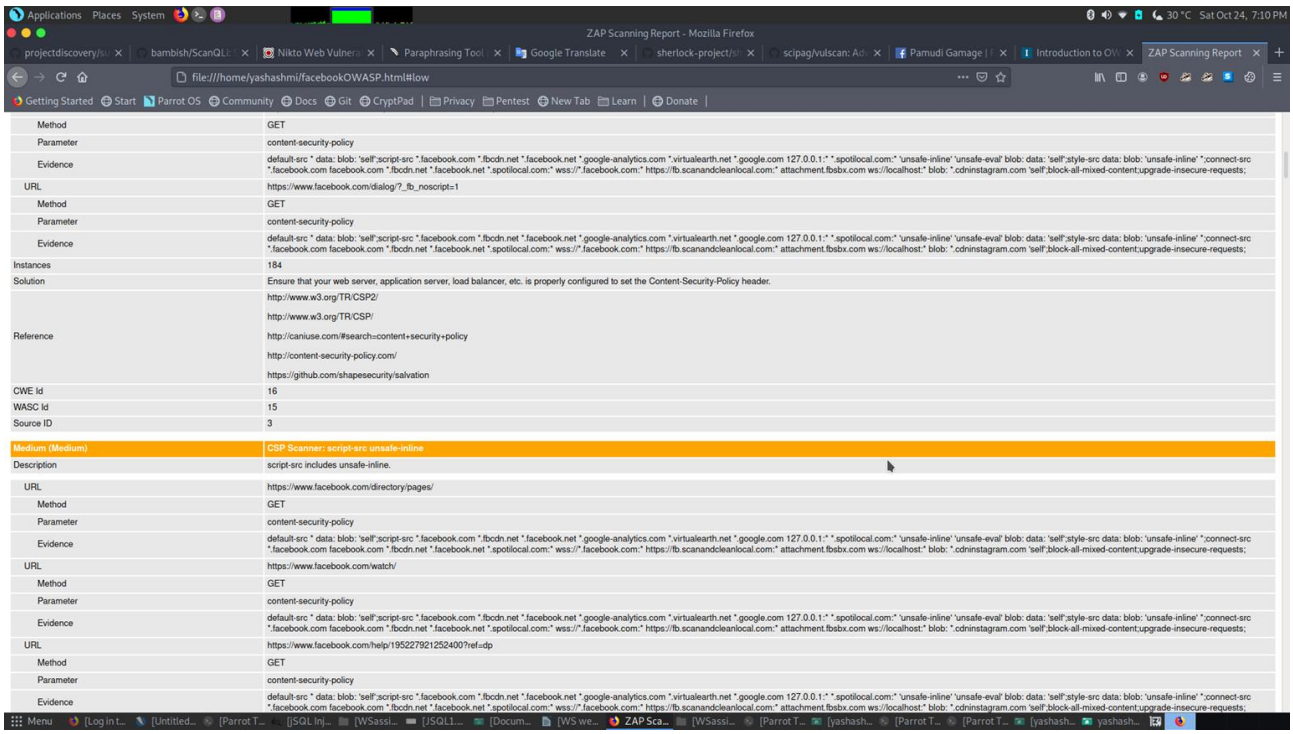
During web application creation, the OWASP ZAP instrument may be used by web developers or by professional security experts during penetration testing to analyze web applications for vulnerabilities.

The OWASP Zed Attack Proxy is a Java-based tool with an intuitive graphical interface that allows web application security testers to conduct fuzzing, scripting, spidering, and proxying to attack web applications.




Applications Places System
30 °C Thu Oct 22, 11:09 PM
Untitled Session - 20201022-172950 - OWASP ZAP 2.9.0 (as superuser)
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start Request Response
Header: Text Body: Large Response
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store, must-revalidate
X-Frame-Options: DENY
X-XSS-Protection: 0
Strict-Transport-Security: max-age=15552000; preload
Very large response body (202,370 bytes) - switch views (using the pulldown currently showing Body: Large Response above) to display.
Be aware that this message may take some time to load.
You can change the minimum message size used for the Large Response view via Options / Display.
History Search Alerts Output Spider
New Scan Progress: 0: https://www.facebook.com 9% Current Scans: 1 URLs Found: 130548 Nodes Added: 4687 Export
URLs Added Nodes Messages
Processed Method URI Flags
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t15.5256-10/s261x260/20776038_199277439767... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/p350x350/122087493_2839688406314622... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/s261x260/122088400_2838817469735049_... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/p168x128/122094386_2838817293068400... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-1/p720x720/67495750_515120195695439_9... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-1/cp0/p56x56/47446254_2347314468629571... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-1/cp0/p50x50/67495750_515120195695439_... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/s480x480/122104019_822777464929709_3... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/p480x480/121990182_821252208415568_... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t15.5256-10/p526x296/73061503_466173883992... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t15.5256-10/s851x315/66517560_402634780377... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t15.5256-10/s261x260/98343271_299425198066... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/p526x296/122104019_822777464929709_... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-9/s851x315/121990182_821252208415568_1... Out of Scope
GET https://scontent.fcmb9-1.fna.fbcdn.net/v/t1.0-0/s261x260/121371736_816456248895164_2... Out of Scope
POST https://www.facebook.com/login/device-based/regular/login/?login_attempt=1&lwv=110
POST https://www.facebook.com/login/device-based/regular/login/?login_attempt=1&lwv=110
POST https://www.facebook.com/login/device-based/regular/login/?login_attempt=1&lwv=110
POST https://www.facebook.com/login/device-based/regular/login/?login_attempt=1&lwv=110
POST https://www.facebook.com/ajax/ufi/modify.php
POST https://www.facebook.com/ajax/ufi/modify.php
POST https://www.facebook.com/ajax/ufi/modify.php
POST https://www.facebook.com/ajax/ufi/modify.php
Alerts 0 4 9 3 Primary Proxy: localhost:8080
Current Scans 0 0 0 55440 0 0 0 1 0 0
Menu Untitled Session - 2020... [Parrot Terminal] BullsEye0 (Jolanda de K...

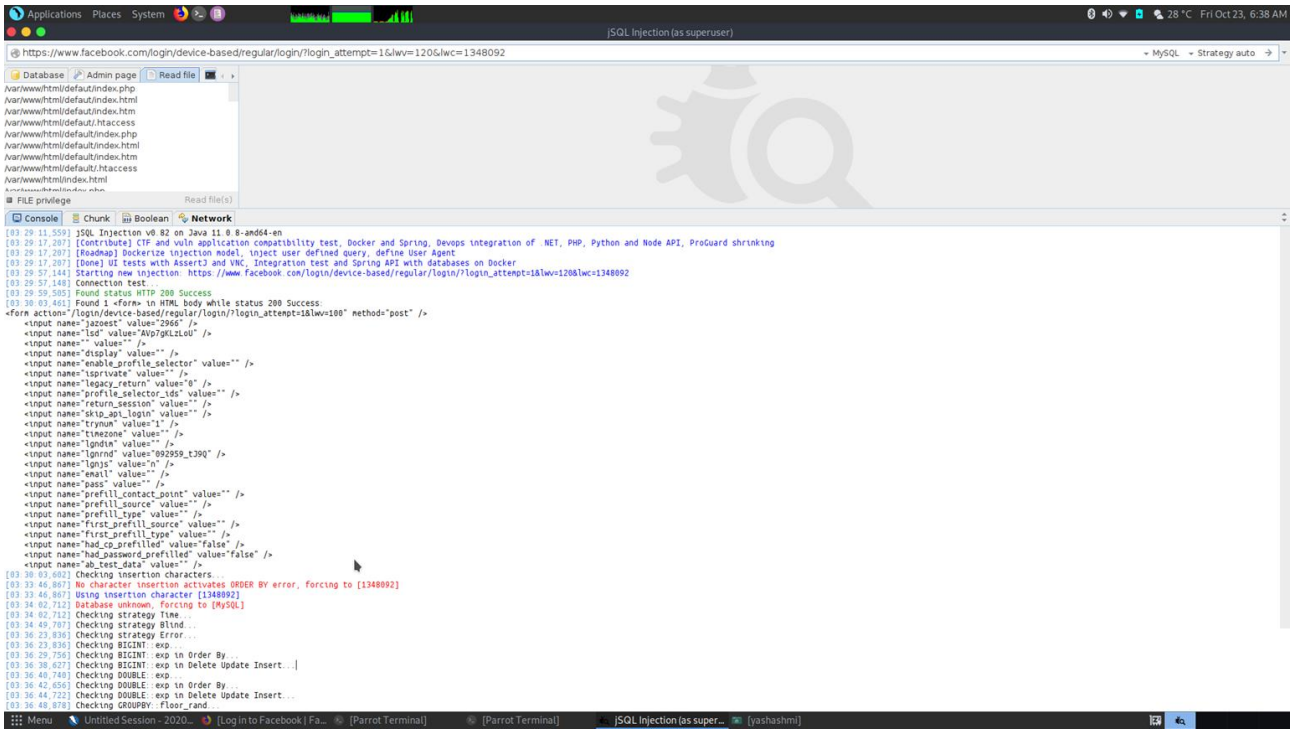

Applications Places System
30 °C Thu Oct 22, 11:09 PM
Untitled Session - 20201022-172950 - OWASP ZAP 2.9.0 (as superuser)
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start Request Response
Header: Text Body: Large Response
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Cache-Control: private, no-cache, no-store, must-revalidate
X-Frame-Options: DENY
X-XSS-Protection: 0
Strict-Transport-Security: max-age=15552000; preload
Very large response body (202,370 bytes) - switch views (using the pulldown currently showing Body: Large Response above) to display.
Be aware that this message may take some time to load.
You can change the minimum message size used for the Large Response view via Options / Display.
History Search Alerts Output Spider
Alerts (16)
CSP Scanner: Wildcard Directive (716)
CSP Scanner: script-src unsafe-inline (715)
CSP Scanner: style-src unsafe-inline (715)
X-Frame-Options Header Not Set (2)
Absence of Anti-CSRF Tokens (1145)
Application Error Disclosure (2)
CSP Scanner: Notices (716)
Cookie No HttpOnly Flag (2)
Cookie Without SameSite Attribute (190)
Cross-Domain JavaScript Source File Inclusion (1455)
Incomplete or No Cache-control and Pragma HTTP Head
Web Browser XSS Protection Not Enabled (703)
X-Content-Type-Options Header Missing (2)
Information Disclosure - Suspicious Comments (726)
Loosely Scoped Cookie (222)
Timestamp Disclosure - Unix (856427)
Web Browser XSS Protection Not Enabled
URL: https://www.facebook.com
Risk: Low
Confidence: Medium
Parameter: X-XSS-Protection
Attack:
Evidence: X-XSS-Protection: 0
CWE ID: 933
WASC ID: 14
Source: Passive (10016 - Web Browser XSS Protection Not Enabled)
Description:
Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server
Other Info:
The X-XSS-Protection HTTP response header allows the web server to enable or disable the web browser's XSS protection mechanism. The following values would attempt to enable it:
X-XSS-Protection: 1; mode=block
X-XSS-Protection: 1; report=http://www.example.com/xss
Solution:
Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.
Reference:
Alerts 0 4 9 3 Primary Proxy: localhost:8080
Current Scans 0 0 0 55501 0 0 0 1 0 0
Menu Untitled Session - 2020... [Parrot Terminal] BullsEye0 (Jolanda de K...

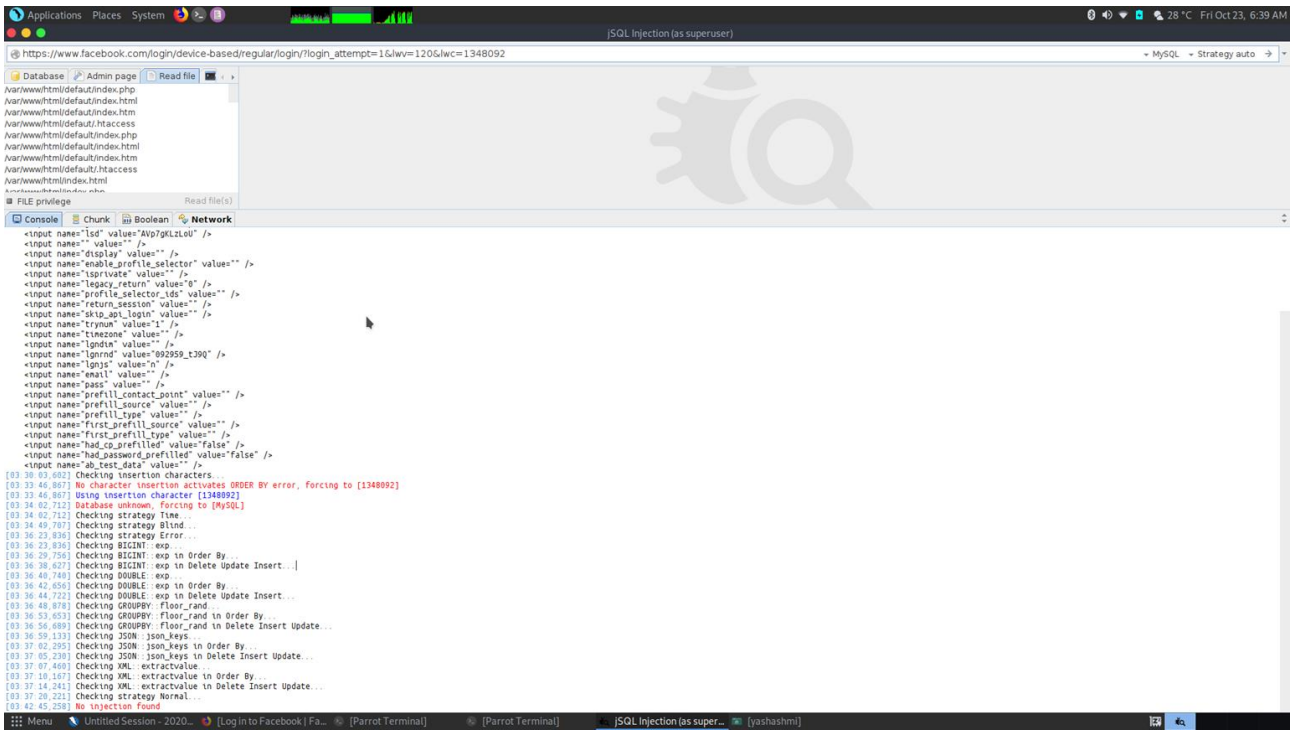


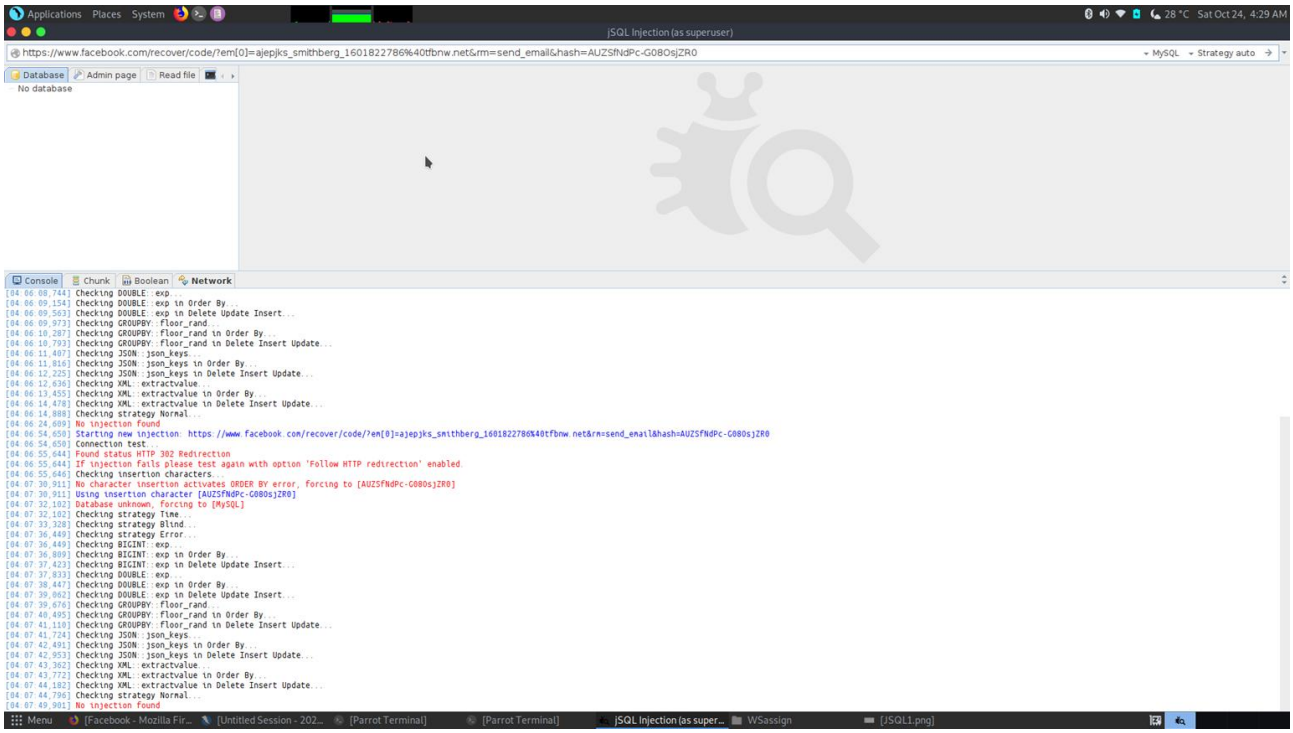
By entering url, can scan the web domain. Issues will show in alerts part. Can create a report also. In that report there descriptions about issues and it will mention solutions for that issues.

JSQL

A multipurpose penetration testing tool is jSQL Injection. Its key role is to identify and exploit vulnerabilities in SQL-injection. This tool was created using java.







To use this tool, have to enter a valid url. I used :

https://www.facebook.com/recover/code/?em[0]=ajepjks_smithberg_1601822786%40tfbnw.net&rm=send_email&hash=AUZSfNdPc-G08Osj3Os

facebook account recovery url.

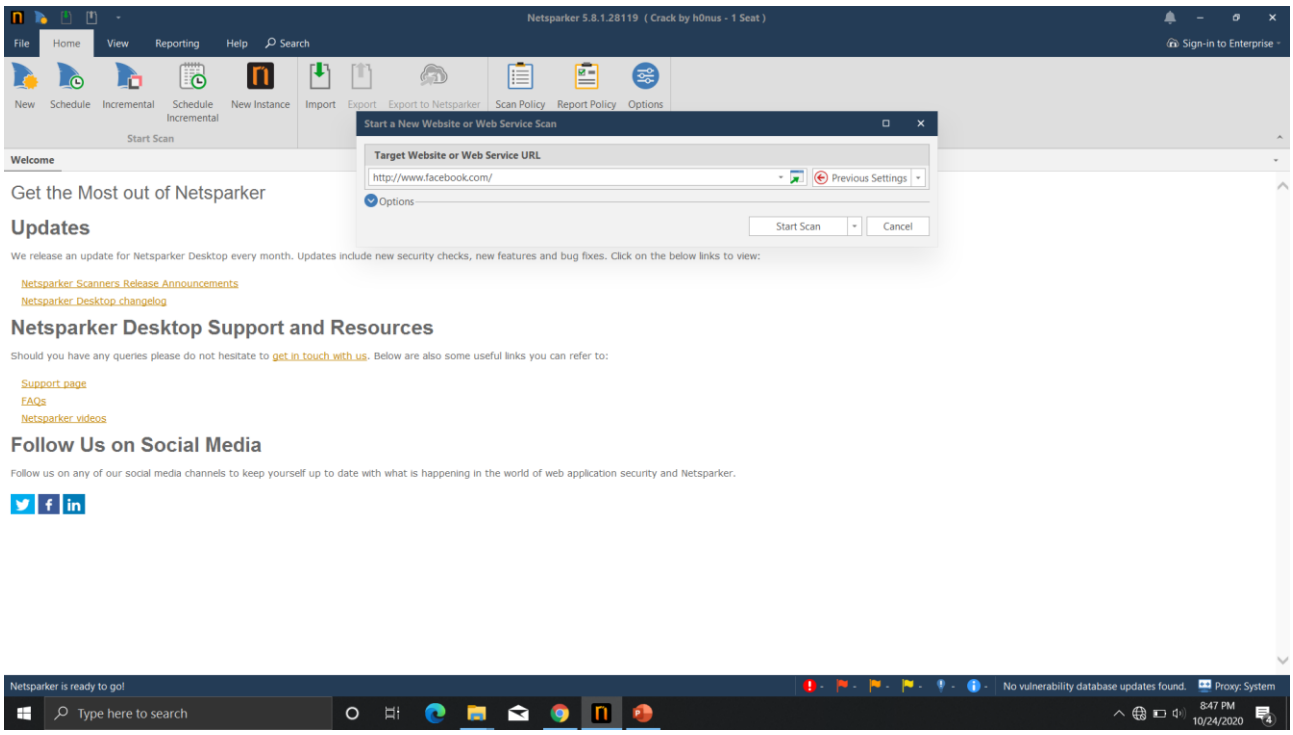
According to jsql tool there were no vulnerabilities found.

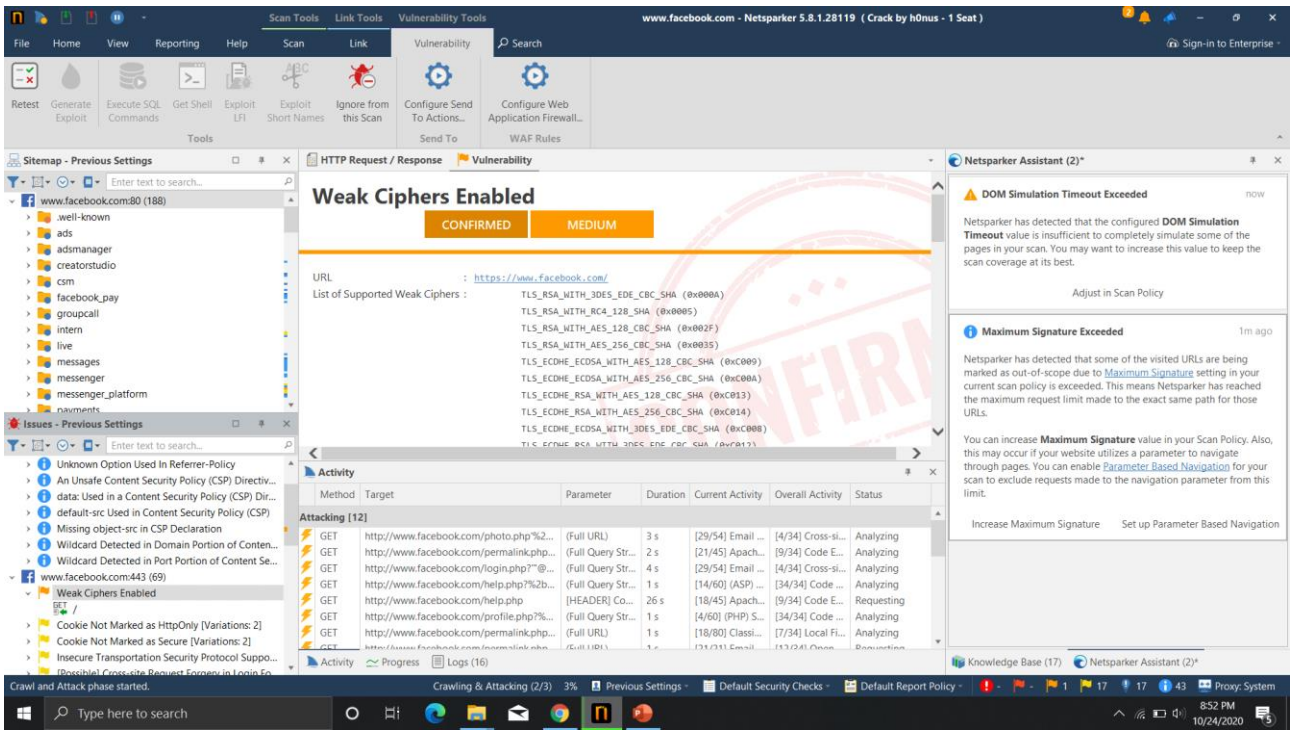
Netsparker

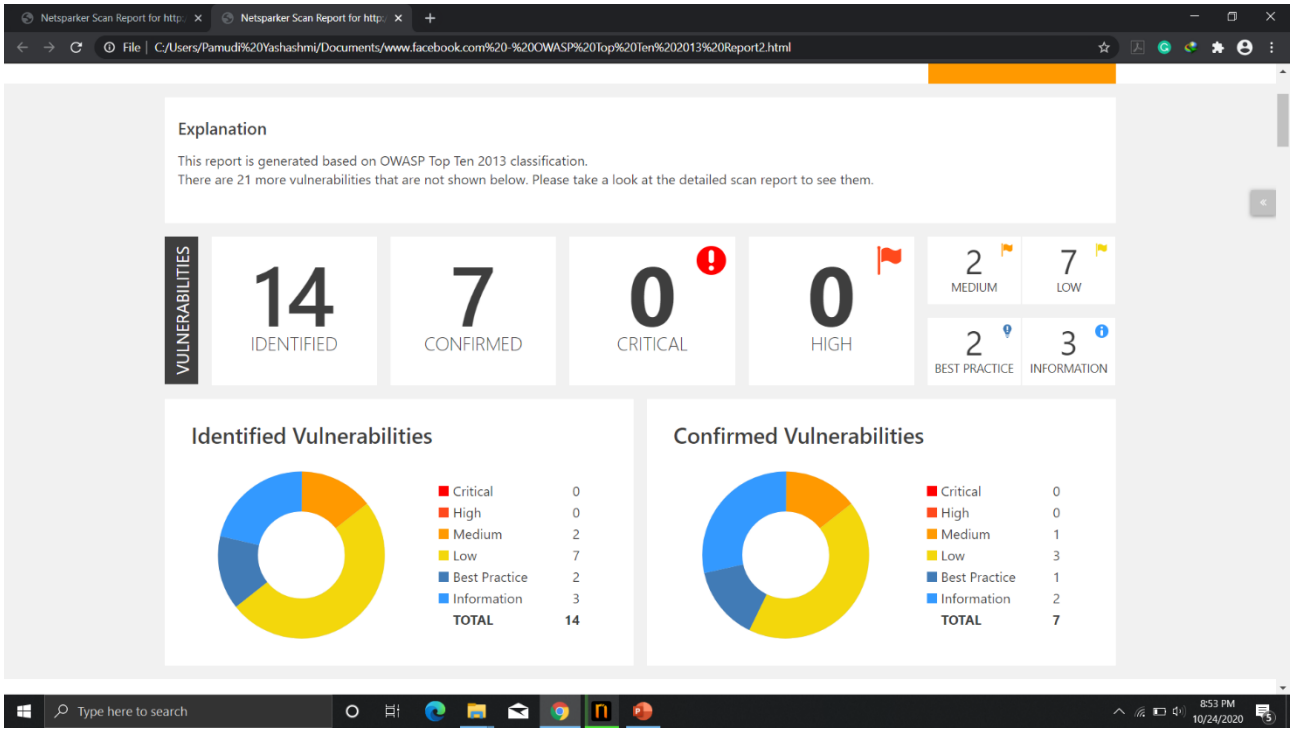
Netsparker is an automatic, but completely configurable, security scanner for web applications that allows you to search websites, web applications and web services and detect security vulnerabilities. All kinds of web applications can be scanned by Netsparker, regardless of the platform or the language in which they are designed.

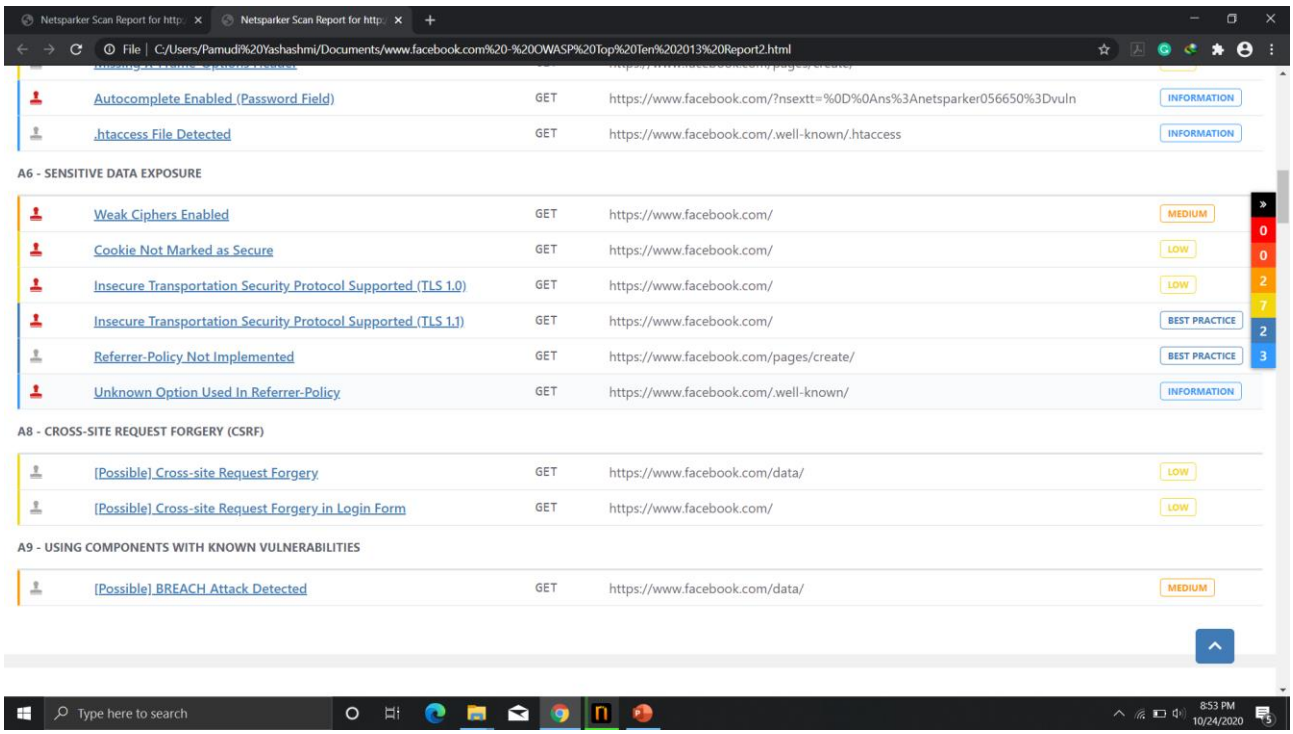
In order to validate known problems, Netsparker is the only online web application security scanner that automatically exploits identified vulnerabilities in a read-only and secure manner. It also offers evidence of the weakness so that you do not have to spend time checking it manually. In the case of a detected SQL injection vulnerability, for example, the database name will be displayed as the exploit proof.

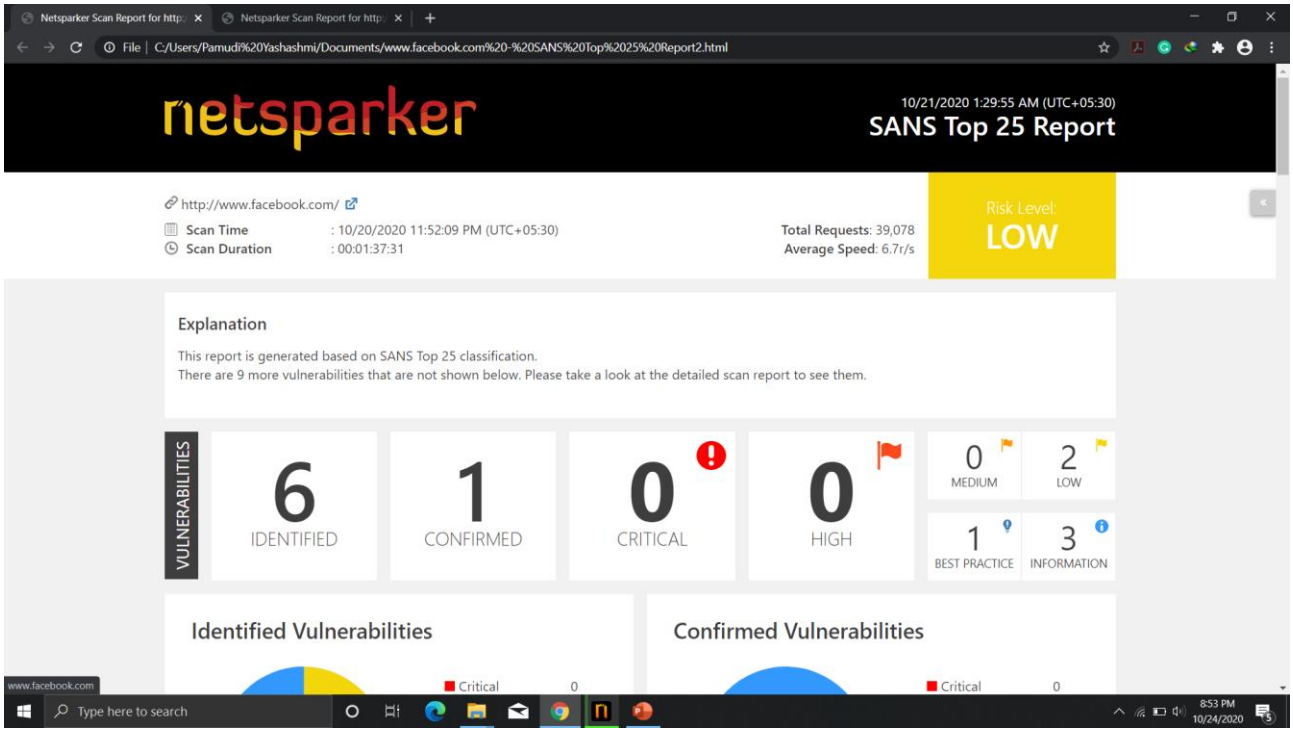
To scan the domain, need to enter the URL. (https://www.facebook.com)

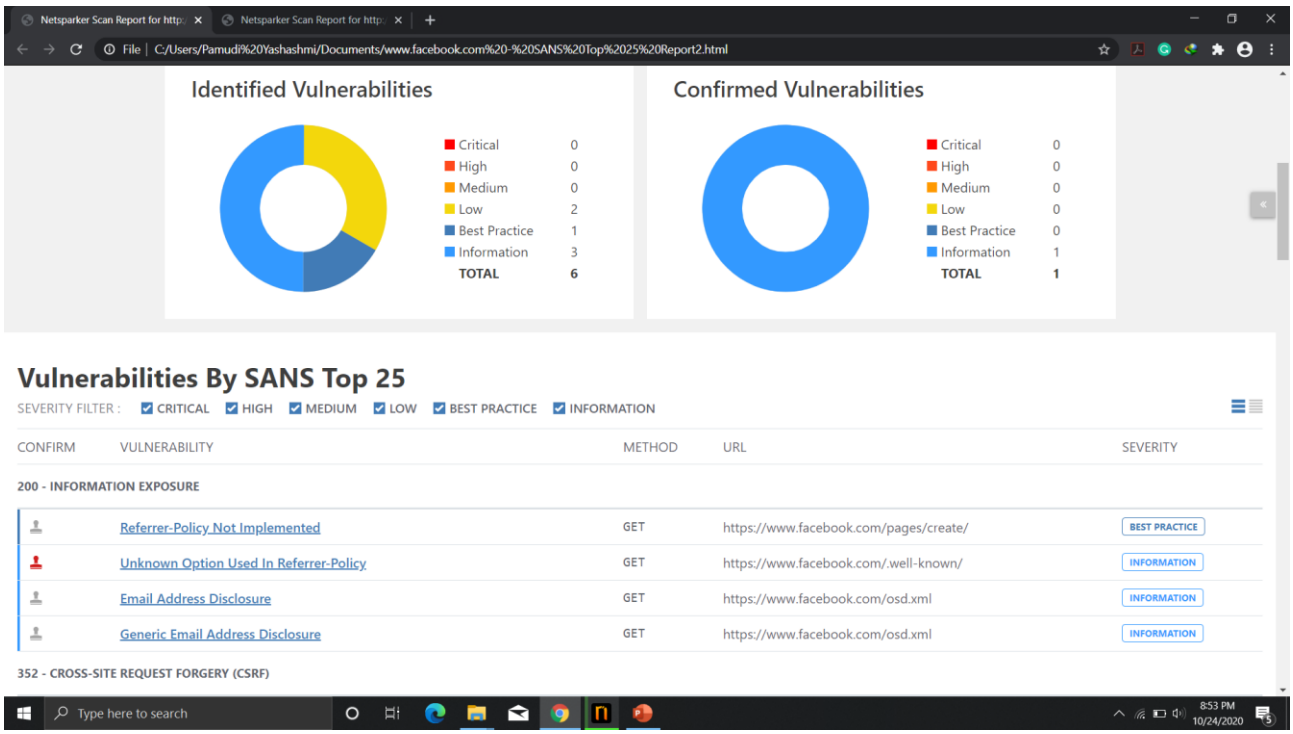


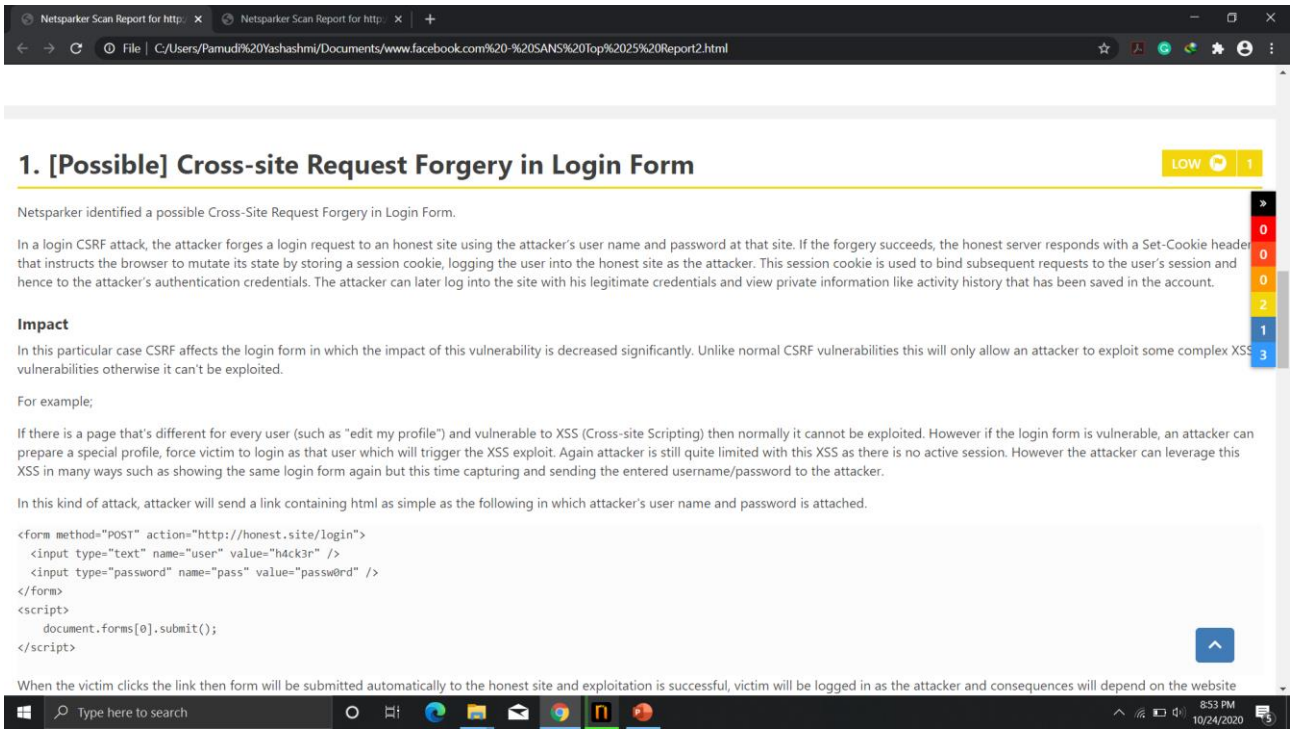

www.facebook.com - Netsparker 5.8.1.28119 (Crack by h0nus - 1 Seat)
Weak Ciphers Enabled
CONFIRMED
MEDIUM
URL : https://www.facebook.com/
List of Supported Weak Ciphers :
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000A)
TLS_RSA_WITH_RC4_128_SHA (0x0005)
TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xC009)
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xC00A)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xC008)
Netsparker Assistant (2)*
DOM Simulation Timeout Exceeded
Netsparker has detected that the configured DOM Simulation Timeout value is insufficient to completely simulate some of the pages in your scan. You may want to increase this value to keep the scan coverage at its best.
Adjust in Scan Policy
Maximum Signature Exceeded
1m ago
Netsparker has detected that some of the visited URLs are being marked as out-of-scope due to Maximum Signature setting in your current scan policy is exceeded. This means Netsparker has reached the maximum request limit made to the exact same path for those URLs.
You can increase Maximum Signature value in your Scan Policy. Also, this may occur if your website utilizes a parameter to navigate through pages. You can enable Parameter Based Navigation for your scan to exclude requests made to the navigation parameter from this limit.
Increase Maximum Signature
Set up Parameter Based Navigation
Activity
Attacking [12]
Crawl and Attack phase started.


Explanation
This report is generated based on OWASP Top Ten 2013 classification.
There are 21 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.
VULNERABILITIES
14 IDENTIFIED
7 CONFIRMED
0 CRITICAL
0 HIGH
2 MEDIUM
7 LOW
2 BEST PRACTICE
3 INFORMATION
Identified Vulnerabilities
Critical 0
High 0
Medium 2
Low 7
Best Practice 2
Information 3
TOTAL 14
Confirmed Vulnerabilities
Critical 0
High 0
Medium 1
Low 3
Best Practice 1
Information 2
TOTAL 7


Netsparker Scan Report for http:
Netsparker Scan Report for http:
File | C:/Users/Pamudi%20Yashashmi/Documents/www.facebook.com%20-%20OWASP%20Top%20Ten%202013%20Report2.html
Autocomplete Enabled (Password Field) GET https://www.facebook.com/?nsextt=%0D%0Ans%3Anetsparker056650%3Dvuln INFORMATION
.htaccess File Detected GET https://www.facebook.com/.well-known/.htaccess INFORMATION
A6 - SENSITIVE DATA EXPOSURE
Weak Ciphers Enabled GET https://www.facebook.com/ MEDIUM
Cookie Not Marked as Secure GET https://www.facebook.com/ LOW
Insecure Transportation Security Protocol Supported (TLS 1.0) GET https://www.facebook.com/ LOW
Insecure Transportation Security Protocol Supported (TLS 1.1) GET https://www.facebook.com/ BEST PRACTICE
Referrer-Policy Not Implemented GET https://www.facebook.com/pages/create/ BEST PRACTICE
Unknown Option Used In Referrer-Policy GET https://www.facebook.com/.well-known/ INFORMATION
A8 - CROSS-SITE REQUEST FORGERY (CSRF)
[Possible] Cross-site Request Forgery GET https://www.facebook.com/data/ LOW
[Possible] Cross-site Request Forgery in Login Form GET https://www.facebook.com/ LOW
A9 - USING COMPONENTS WITH KNOWN VULNERABILITIES
[Possible] BREACH Attack Detected GET https://www.facebook.com/data/ MEDIUM
0
0
2
7
2
3
Type here to search
8:53 PM
10/24/2020


Netsparker Scan Report for http:
Netsparker Scan Report for http:
File | C:/Users/Pamudi%20Yashashmi/Documents/www.facebook.com%20-%20SANS%20Top%2025%20Report2.html
netsparker
10/21/2020 1:29:55 AM (UTC+05:30)
SANS Top 25 Report
http://www.facebook.com/
Scan Time : 10/20/2020 11:52:09 PM (UTC+05:30)
Scan Duration : 00:01:37:31
Total Requests: 39,078
Average Speed: 6.7r/s
Risk Level:
LOW
Explanation
This report is generated based on SANS Top 25 classification.
There are 9 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.
VULNERABILITIES
6 IDENTIFIED
1 CONFIRMED
0 CRITICAL
0 HIGH
0 MEDIUM
2 LOW
1 BEST PRACTICE
3 INFORMATION
Identified Vulnerabilities
Confirmed Vulnerabilities
Critical 0
Critical 0
www.facebook.com
Type here to search
8:53 PM
10/24/2020


Netsparker Scan Report for http:
Netsparker Scan Report for http:
File | C:/Users/Pamudi%20Yashashmi/Documents/www.facebook.com%20-%20SANS%20Top%2025%20Report2.html
Identified Vulnerabilities
Critical 0
High 0
Medium 0
Low 2
Best Practice 1
Information 3
TOTAL 6
Confirmed Vulnerabilities
Critical 0
High 0
Medium 0
Low 0
Best Practice 0
Information 1
TOTAL 1
Vulnerabilities By SANS Top 25
SEVERITY FILTER : CRITICAL HIGH MEDIUM LOW BEST PRACTICE INFORMATION
CONFIRM VULNERABILITY METHOD URL SEVERITY
200 - INFORMATION EXPOSURE
Referrer-Policy Not Implemented GET https://www.facebook.com/pages/create/ BEST PRACTICE
Unknown Option Used In Referrer-Policy GET https://www.facebook.com/.well-known/ INFORMATION
Email Address Disclosure GET https://www.facebook.com/osd.xml INFORMATION
Generic Email Address Disclosure GET https://www.facebook.com/osd.xml INFORMATION
352 - CROSS-SITE REQUEST FORGERY (CSRF)
Type here to search
8:53 PM
10/24/2020


Netsparker Scan Report for http:
Netsparker Scan Report for http:
File | C:/Users/Pamudi%20Yashashmi/Documents/www.facebook.com%20-%20SANS%20Top%2025%20Report2.html
1. [Possible] Cross-site Request Forgery in Login Form
LOW 1
Netsparker identified a possible Cross-Site Request Forgery in Login Form.
In a login CSRF attack, the attacker forges a login request to an honest site using the attacker's user name and password at that site. If the forgery succeeds, the honest server responds with a Set-Cookie header that instructs the browser to mutate its state by storing a session cookie, logging the user into the honest site as the attacker. This session cookie is used to bind subsequent requests to the user's session and hence to the attacker's authentication credentials. The attacker can later log into the site with his legitimate credentials and view private information like activity history that has been saved in the account.
Impact
In this particular case CSRF affects the login form in which the impact of this vulnerability is decreased significantly. Unlike normal CSRF vulnerabilities this will only allow an attacker to exploit some complex XSS vulnerabilities otherwise it can't be exploited.
For example;
If there is a page that's different for every user (such as "edit my profile") and vulnerable to XSS (Cross-site Scripting) then normally it cannot be exploited. However if the login form is vulnerable, an attacker can prepare a special profile, force victim to login as that user which will trigger the XSS exploit. Again attacker is still quite limited with this XSS as there is no active session. However the attacker can leverage this XSS in many ways such as showing the same login form again but this time capturing and sending the entered username/password to the attacker.
In this kind of attack, attacker will send a link containing html as simple as the following in which attacker's user name and password is attached.
<form method="POST" action="http://honest.site/login">
<input type="text" name="user" value="h4ck3r" />
<input type="password" name="pass" value="passw0rd" />
</form>
<script>
document.forms[0].submit();
</script>
When the victim clicks the link then form will be submitted automatically to the honest site and exploitation is successful, victim will be logged in as the attacker and consequences will depend on the website
Type here to search
8:53 PM
10/24/2020

Vulnerabilities and Mitigation Methods

1. BREACH attack vulnerability (https://www.facebook.com/login/)

BREACH attacks are attacks similar to the CRIME attack, abbreviated from Browser Reconnaissance and Ex-filtration via Adaptive Compression of Hypertext. Both attacks are side-channel compression attacks, but CRIME targets information compressed through TLS compression in HTTP requests, while BREACH targets information compressed through HTTP compression in HTTP responses. The deflate algorithm, which is a data compression algorithm that is made up of a combination of Huffman coding and LZ77 compression, typically performs HTTP compression. Any repeated byte sequences in the input are observed when compressing data using this algorithm and are not repeated in the output. Instead, along with pointers pointing out where the same sequence is located again, the repeated byte sequence is stored only once. This will decrease the amount of bytes being sent, thus also reducing the time it takes to send the information. However, the length of the compressed data is still noticeable even when encrypted, and this is one of the essential elements that makes the BREACH attack feasible. In addition, it must be served from a server using HTTP compression in order for an application to be vulnerable to a BREACH attack, and it must also contain user-input and a secret in the HTTP response body, such as a CSRF token. An attacker exploiting a vulnerability to a BREACH attack will need to have a way to view the traffic of the victim and also have the ability to allow the victim to send HTTP requests to the compromised website, which could be achieved by persuading the victim to access an attacker-controlled malicious site. This site will be built in a manner that does not suspect the end of the victim.

Can mitigate this by:

- Separating secrets from user input
- Randomizing secrets per request
- Masking secrets (effectively randomizing by XORing with a random secret per request)
- Protecting vulnerable pages with CSRF
- Length hiding (by adding random number of bytes to the responses)
- Rate-limiting the requests
- Disabling HTTP compression

2. OSVDB – 23654/login/profile.php? u=dNytSUYE: Powerboards is vulnerable to Path disclosure(https://www.facebook.com/login/)

Full Path Disclosure (FPD) vulnerabilities cause the webroot / file path to be seen by the intruder. Such bugs, such as using the load file) (query to access the page source (within a SQL injection), enable the attacker to provide the entire path to the file they want to access.

Can Mitigate By:

- Error Handling
- Bounds Checking
- Safe Libraries
- Static Code Analysis
- Executable space protection
- Address space layout randomization(ASLR)
- Stack-smashing Protection(SSP)

3. SQL vulnerability found (https://www.facebook.com/login/)

Using scanQLi tool, found it.

4. CSP Scanner: style-src unsafe-inline

(https://www.facebook.com/login.php?display=popup&next=https%3A%2F%2Fwww.facebook.com%2Fshare%2F)

Style-src includes unsafe-inline

To Mitigate that:

- Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

4. CSP Scanner: Wildcard Directive

(https://www.facebook.com/?_fb_noscript=1)

The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined:

style-src, style-src-elem, style-src-attr, img-src, frame-src, frame-ancestor, font-src, media-src, object-src, manifest-src, prefetch-src

To Mitigate that:

- Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

5. X-Frame-Options Header Not Set

(https://www.facebook.com/pages/category/)

X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

To Mitigate that:

- Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).

6. Web Browser XSS Protection Not Enabled

(https://www.facebook.com/photo.php?_fb_noscript=1)

Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server.

To Mitigate This:

- Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.

7. Absence of Anti-CSRF Tokens

(https://www.facebook.com/r.php?display=page&locale=en_GB)

No Anti-CSRF tokens were found in a HTML submission form.

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

CSRF attacks are effective in a number of situations, including:

* The victim has an active session on the target site.

* The victim is authenticated via HTTP auth on the target site.

* The victim is on the same local network as the target site.

CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

To Mitigate This:

- Phase: Architecture and Design

  Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

  For example, use anti-CSRF packages such as the OWASP CSRFGuard.

  Phase: Implementation

  Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.

  Phase: Architecture and Design

  Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).

  Note that this can be bypassed using XSS.

  Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

  Note that this can be bypassed using XSS.

  Use the ESAPI Session Management control.

  This control includes a component for CSRF.

Do not use the GET method for any request that triggers a state change.

  Phase: Implementation

  Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

8. Cookie Without SameSite Attribute

(https://www.facebook.com/hashtag/?locale2=en_GB)

A cookie has been set with an invalid SameSite attribute value, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

To Mitigate This:

- Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

9. Cross-Domain JavaScript Source File Inclusion

(https://www.facebook.com/feeds/?locale2=en_GB)

The page includes one or more script files from a third-party domain.

To Mitigate This:

- Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

10. Application Error Disclosure

(https://www.facebook.com/pages/create/?ref_type=registration_form)

This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.

To Mitigate This:

- Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.

11. Cookie No HttpOnly Flag

(https://www.facebook.com/login/?_fb_noscript=1)

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

To Mitigate This:

- Ensure that the HttpOnly flag is set for all cookies.

12. Incomplete or No Cache-control and Pragma HTTP Header Set

(https://www.facebook.com/pages/category/)

The cache-control and pragma HTTP header have not been set properly or are missing allowing the browser and proxies to cache content.

To Mitigate This:

- Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate; and that the pragma HTTP header is set with no-cache.

13. X-Content-Type-Options Header Missing

(https://www.facebook.com/pages/category/)

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than

the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

To Mitigate This:

- Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.
- If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Conclusion

I selected the facebook.com domain to do the audit. Before do the auditing I created a facebook test account. I used some tools to scan the domain. I mainly focused on SQL injection vulnerabilities. Also I found another vulnerabilities too. Sublist3r, subfinder, sherlock, scanQLi, nikto, vulscan, Netsparker, OWASP ZAP, jSQL are the tools that I used for auditing. At the end I found several vulnerabilities.