Practical 6

Aim: Practical on Exploiting Web-based applications

1. Reconnaissance and Identification of Web applications

Run the following commands to make sure that you Kali Linux distribution is up to date.

a) sudo apt update
b) sudo apt upgrade
c) sudo apt dist-upgrade

The process of waf detection can be automated using nmap script http-waf-detect.nse as shown below:

```
└─$ sudo apt-get update
[sudo] password for kali:
Hit:1 https://packages.sury.org/php buster InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:3 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Packages [19.4 MB]
Get:5 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,082 B]
Get:6 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [45.6 MB]
Get:7 http://kali.download/kali kali-rolling/contrib amd64 Packages [121 kB]
Get:8 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [294 kB]
Get:9 http://kali.download/kali kali-rolling/non-free amd64 Packages [226 kB]
Get:10 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [914 kB]
Fetched 66.6 MB in 17s (3,912 kB/s)
Reading package lists... Done

┌──(kali㉿kali)-[~]
└─$ sudo nmap -vv -p 80 --script http-waf-detect www.testfire.net
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-14 01:04 EST
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 01:04
Completed NSE at 01:04, 0.00s elapsed
Initiating Ping Scan at 01:04
Scanning www.testfire.net (65.61.137.117) [4 ports]
Completed Ping Scan at 01:04, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:04
Completed Parallel DNS resolution of 1 host. at 01:04, 0.36s elapsed
Initiating SYN Stealth Scan at 01:04
Scanning www.testfire.net (65.61.137.117) [1 port]
Completed SYN Stealth Scan at 01:04, 0.23s elapsed (1 total ports)
NSE: Script scanning 65.61.137.117.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 01:04
Completed NSE at 01:04, 0.00s elapsed
Nmap scan report for www.testfire.net (65.61.137.117)
Host is up, received reset ttl 128 (0.0012s latency).
Scanned at 2023-12-14 01:04:39 EST for 0s

PORT   STATE    SERVICE REASON
80/tcp filtered http    no-response

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 01:04
Completed NSE at 01:04, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.90 seconds
           Raw packets sent: 6 (240B) | Rcvd: 1 (40B)
```

Then we will run the python tool WAFW00F to perform the identification and fingerprinting of a Web Application Firewall. In this case we will check the firewall of www.hdfcbank.com.

```
┌──(kali㉿kali)-[~]
└─$ sudo wafw00f www.hdfcbank.com

                ~ WAFW00F : v2.2.0 ~
    The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://www.hdfcbank.com
[+] The site https://www.hdfcbank.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
```

Then we will use a Load Balancing Detector on www.hdfcbank.com.

```
┌──(kali㉿kali)-[~]
└─$ sudo lbd www.hdfcbank.com

lbd - load balancing detector 0.4 - Checks if a given domain uses load-balancing.
                                    Written by Stefan Behte (http://ge.mine.nu)
                                    Proof-of-concept! Might give false positives.

Checking for DNS-Loadbalancing: FOUND
www.hdfcbank.com has address 104.17.6.56
www.hdfcbank.com has address 104.16.36.67

Checking for HTTP-Loadbalancing [Server]:
 cloudflare
 NOT FOUND

Checking for HTTP-Loadbalancing [Date]: 06:06:20, 06:06:20, 06:06:21, 06:06:21, 06:06:21,
06:06:21, 06:06:21, 06:06:22, 06:06:22, 06:06:22, 06:06:22, 06:06:22, 06:06:22, 06:06:22,
06:06:23, 06:06:23, 06:06:23, 06:06:23, 06:06:23, 06:06:23, 06:06:23, 06:06:23, 06:06:23,

Checking for HTTP-Loadbalancing [Diff]: FOUND
< Expires: Thu, 14 Dec 2023 07:06:23 GMT
> Expires: Thu, 14 Dec 2023 07:06:24 GMT
< CF-RAY: 835442b7dbe91bd6-BOM
> CF-RAY: 835442b81d988489-BOM

www.hdfcbank.com does Load-balancing. Found via Methods: DNS HTTP[Diff]
```

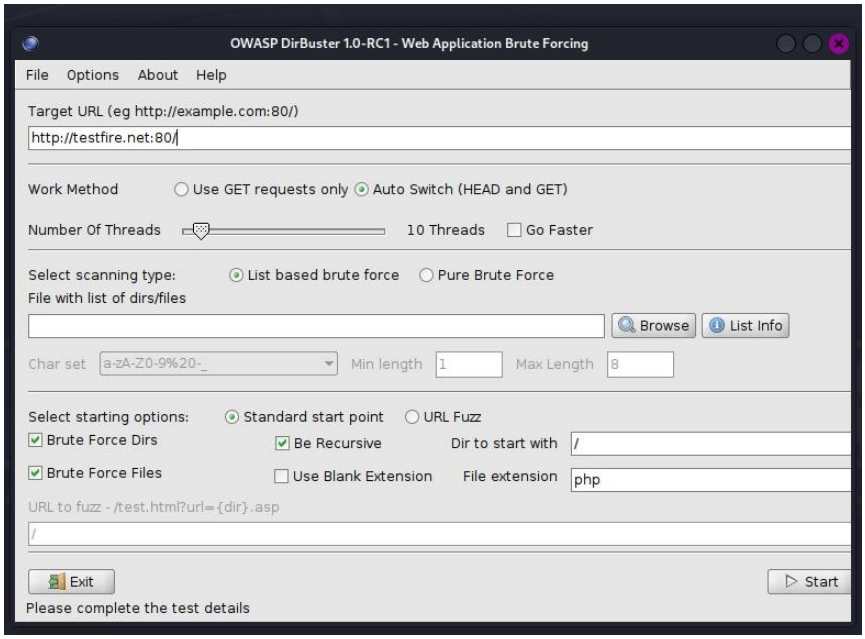
After that, we will perform a WordPress scan on blogs.overandall.com to check for any WordPress vulnerabilities that we can exploit.

```
┌──(kali㉿kali)-[~]
└─$ sudo wpscan --url blogs.overandall.com
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

      WordPress Security Scanner by the WPScan Team
                      Version 3.8.25
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: https://blogs.overandall.com/ [104.21.8.216]
[+] Started: Thu Dec 14 01:10:40 2023

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - x-powered-by: PHP/7.4.33
 |  - x-litespeed-cache: hit
 |  - platform: hostinger
 |  - x-turbo-charged-by: LiteSpeed
 |  - cf-cache-status: DYNAMIC
 |  - server: cloudflare
 |  - cf-ray: 835448837aee6ebc-BOM
 |  - alt-svc: h3=":443"; ma=86400
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

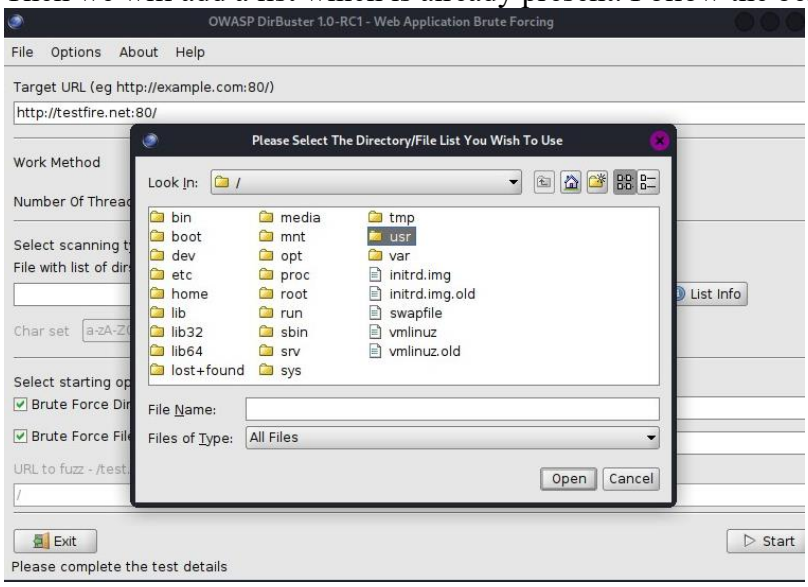
```
[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Thu Dec 14 01:10:49 2023
[+] Requests Done: 139
[+] Cached Requests: 39
[+] Data Sent: 36.982 KB
[+] Data Received: 54.97 KB
[+] Memory used: 259.383 MB
[+] Elapsed time: 00:00:09

┌──(kali㉿kali)-[~]
└─$
ove the mouse pointer inside or press Ctrl+G.
```

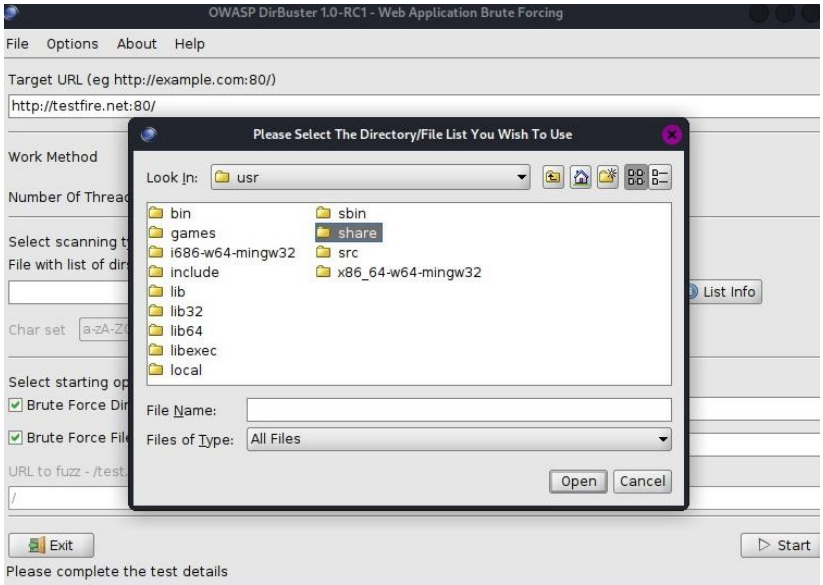
Then we will use the OWASP directory buster to brute force our way through the target website to get the websites directory structure. To use the OWASP directory buster, you can use the following steps. Here our target website will be “www.testfire.net:80/”.

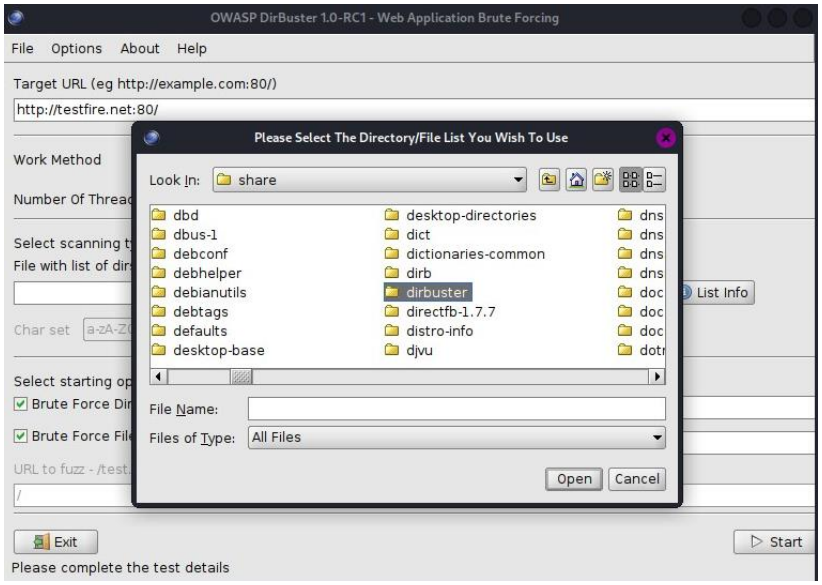
To start dirbuster just type **‘sudo dirbuster’** in command line.

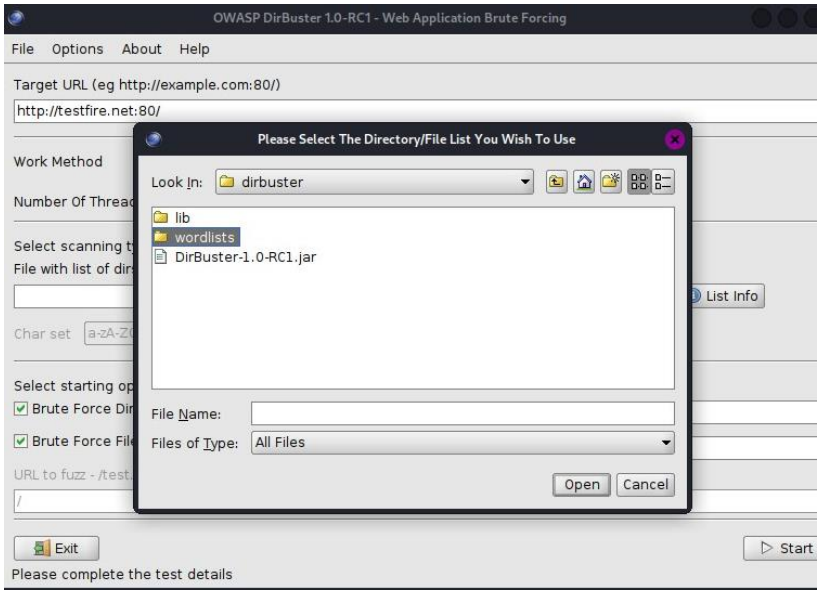


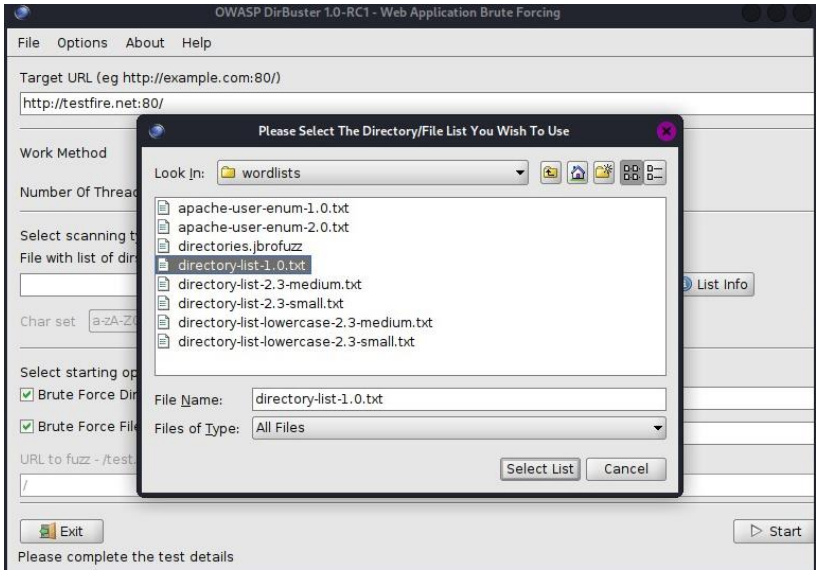
Then we will add a list which is already present. Follow the below steps:

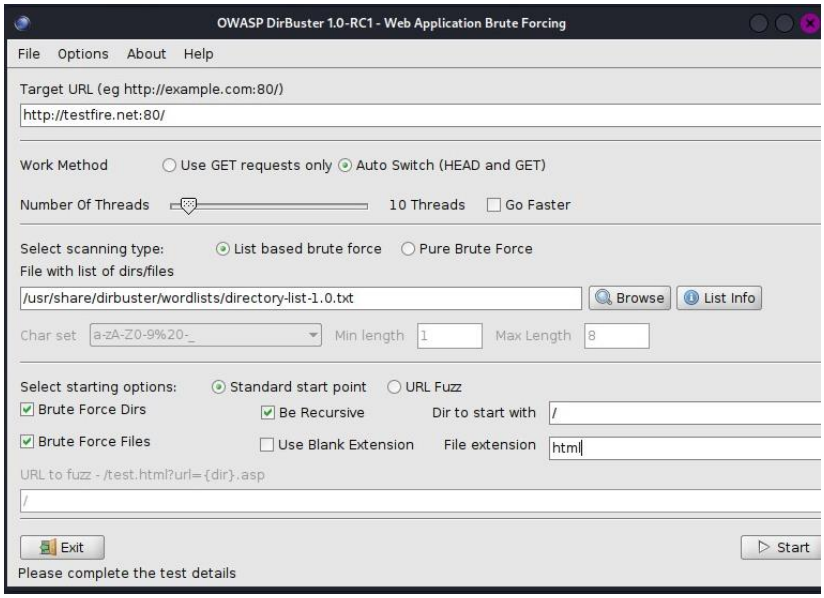




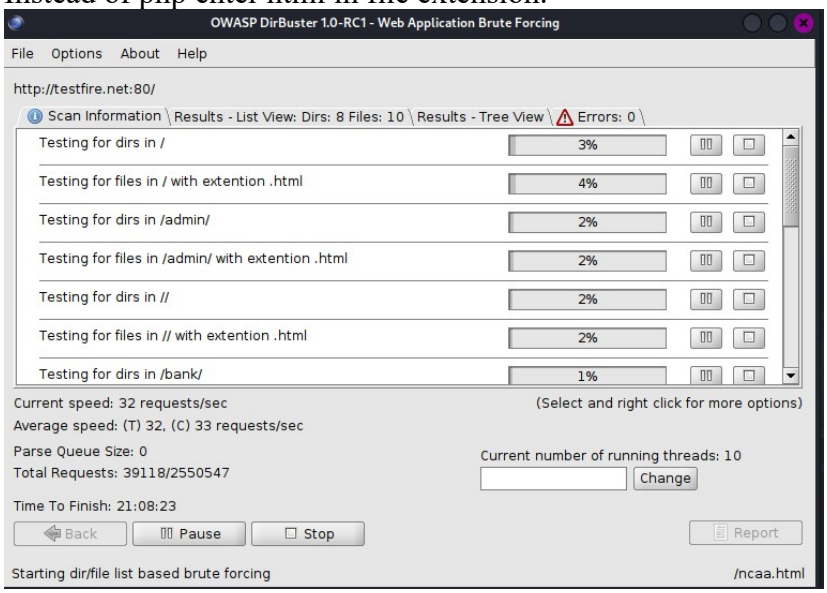

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File Options About Help
Target URL (eg http://example.com:80/)
http://testfire.net:80/
Work Method
Number Of Thread
Select scanning t
File with list of dir
List Info
Char set
Select starting op
Brute Force Dir
Brute Force File
URL to fuzz - /test
Exit
Start
Please complete the test details
Please Select The Directory/File List You Wish To Use
Look In: share
dbd
dbus-1
debconf
debhelper
debianutils
debtags
defaults
desktop-base
desktop-directories
dict
dictionaries-common
dirb
dirbuster
directfb-1.7.7
distro-info
djvu
dns
dns
dns
dns
doc
doc
doc
File Name:
Files of Type: All Files
Open
Cancel

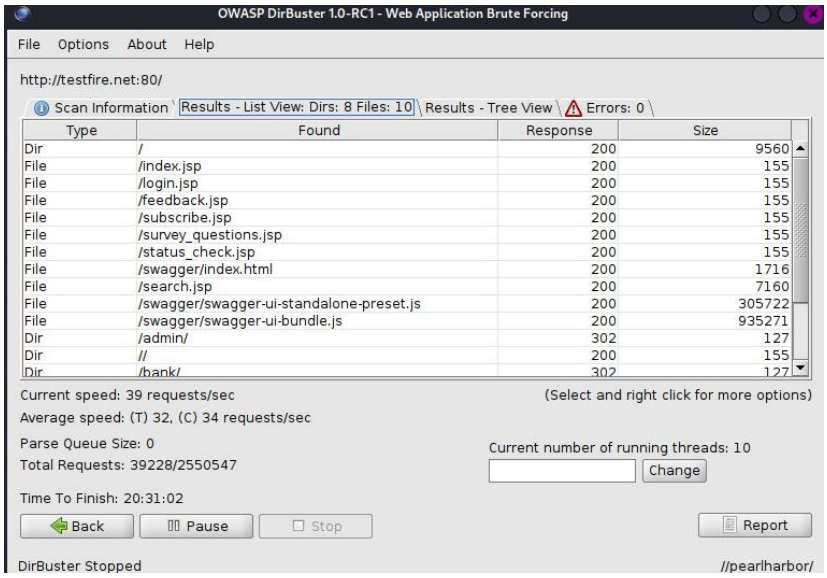

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File Options About Help
Target URL (eg http://example.com:80/)
http://testfire.net:80/
Work Method
Number Of Thread
Select scanning t
File with list of dir
List Info
Char set
Select starting op
Brute Force Dir
Brute Force File
URL to fuzz - /test
Exit
Start
Please complete the test details
Please Select The Directory/File List You Wish To Use
Look In: dirbuster
lib
wordlists
DirBuster-1.0-RC1.jar
File Name:
Files of Type: All Files
Open
Cancel

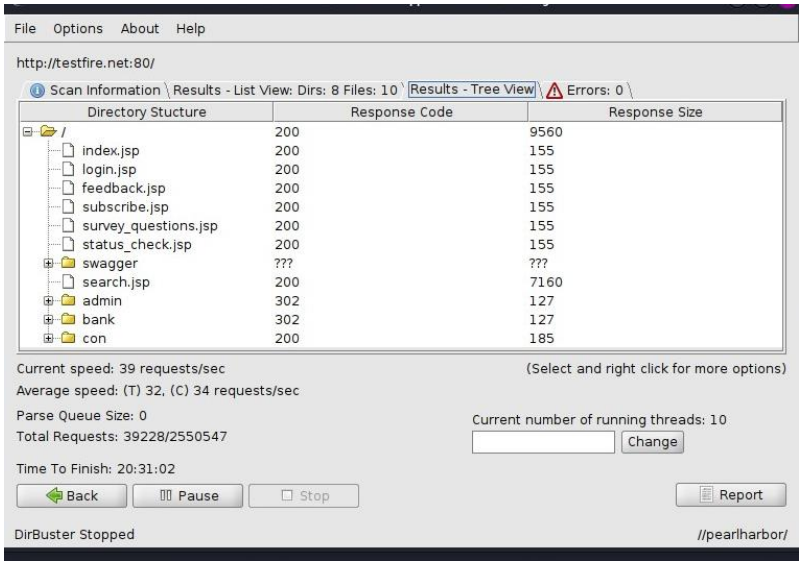

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File Options About Help
Target URL (eg http://example.com:80/)
http://testfire.net:80/
Work Method
Number Of Thread
Select scanning t
File with list of dir
List Info
Char set
Select starting op
Brute Force Dir
Brute Force File
URL to fuzz - /test
Exit
Start
Please complete the test details
Please Select The Directory/File List You Wish To Use
Look In: wordlists
apache-user-enum-1.0.txt
apache-user-enum-2.0.txt
directories.jbrofuzz
directory-list-1.0.txt
directory-list-2.3-medium.txt
directory-list-2.3-small.txt
directory-list-lowercase-2.3-medium.txt
directory-list-lowercase-2.3-small.txt
File Name: directory-list-1.0.txt
Files of Type: All Files
Select List
Cancel



Instead of php enter html in file extension.

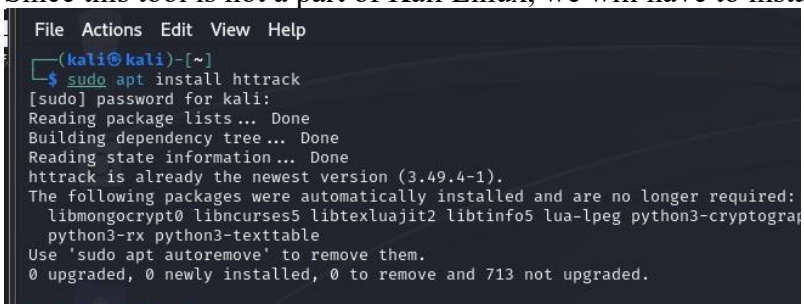




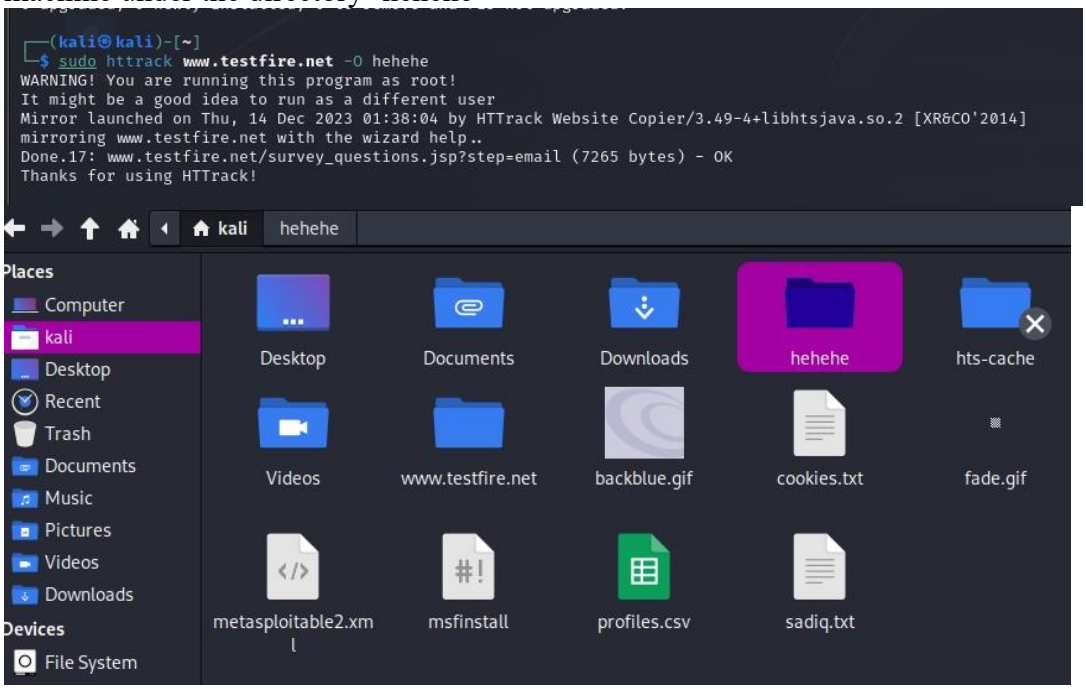


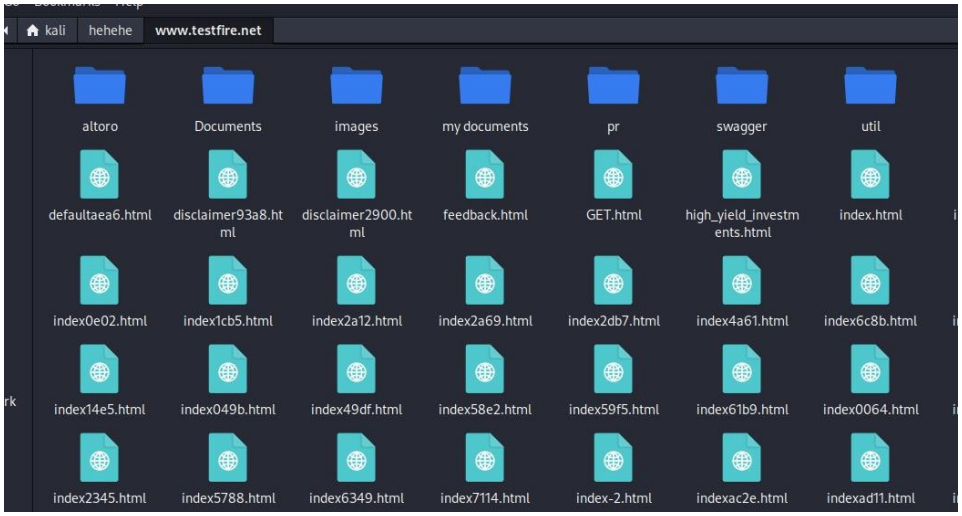
2. Mirroring a website from the command line

Here we will use HTTRACK, which is an open-source web-crawler that can completely clone a website along with all its directories and its overall file structure.
Since this tool is not a part of Kali Linux, we will have to install it.



Then we will copy our target website www.testfire.net by using this tool and will save it on our machine under the directory ‘hehehe’





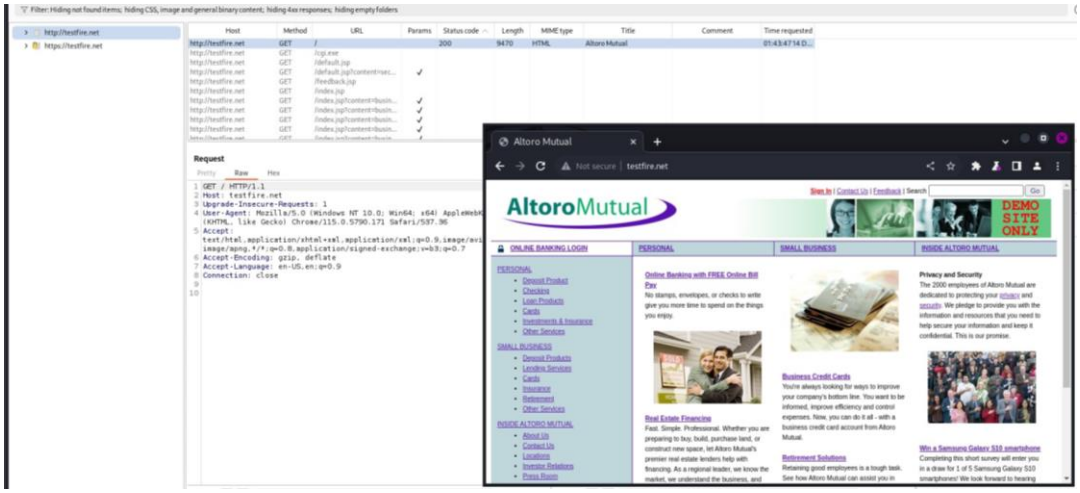
3. Now will use Burp Suite to perform reconnaissance and exploits.

We can access it in the start window of Kali Linux since it comes pre-installed.

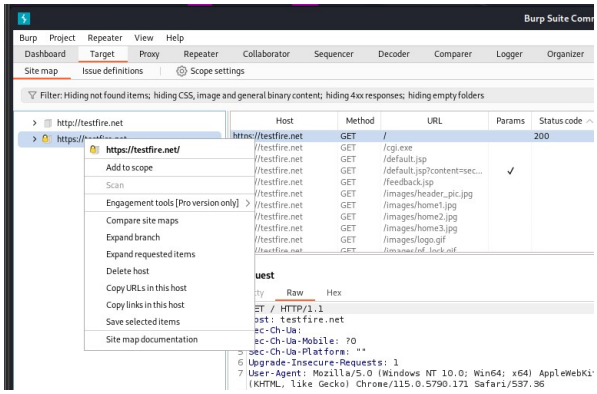
Next, we will create a temporary project.

Then will perform a passive crawl through our target website. Here our target website is “www.testfire.net”. To perform the passive crawl, we have to navigate to the “Target” sub-menu access the in-built browser on the “Sitemap”. We enter our target website into the in-built browser.

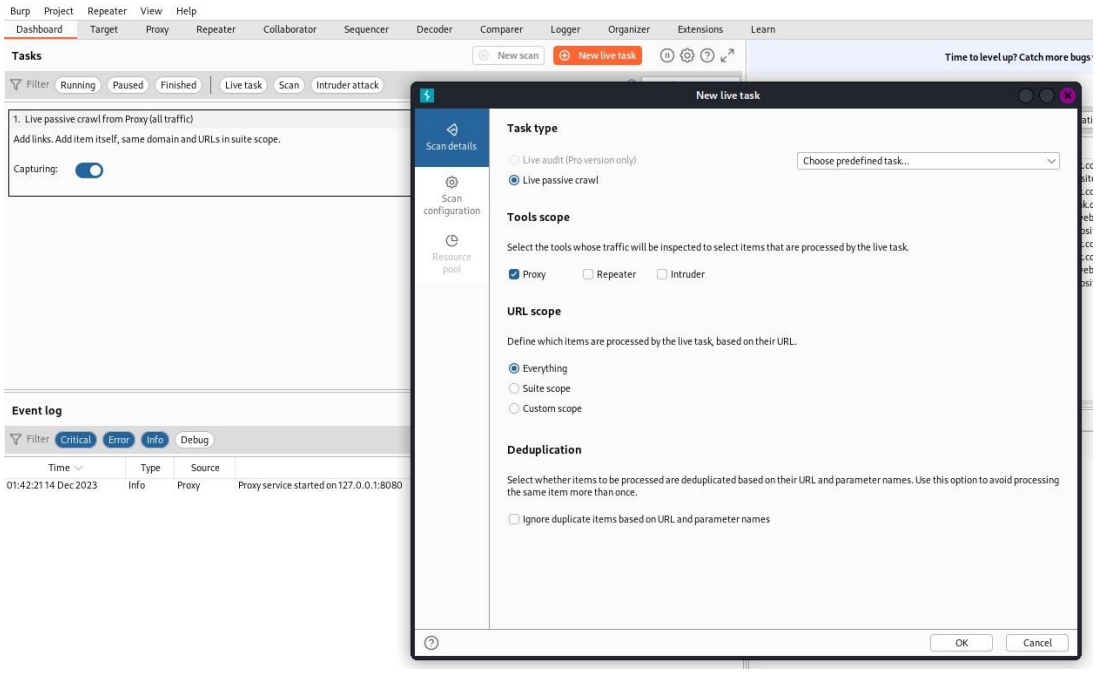
We will start to see the traffic, or the requests being issued on the website in the SiteMap.

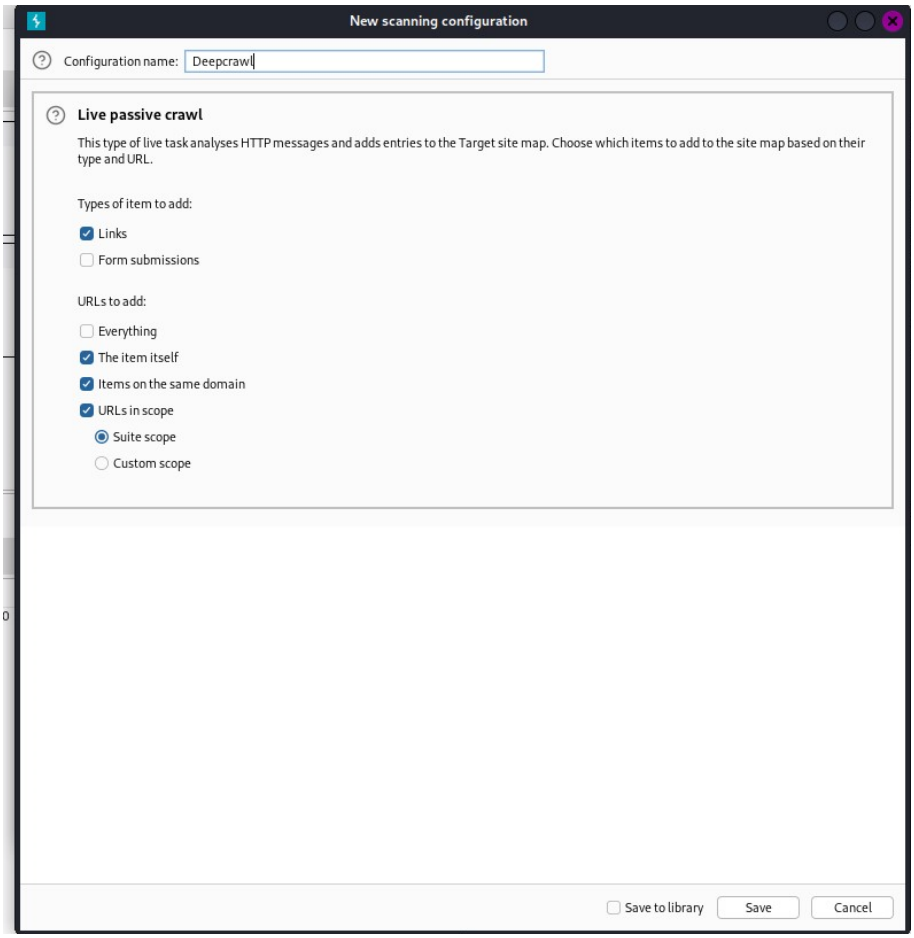


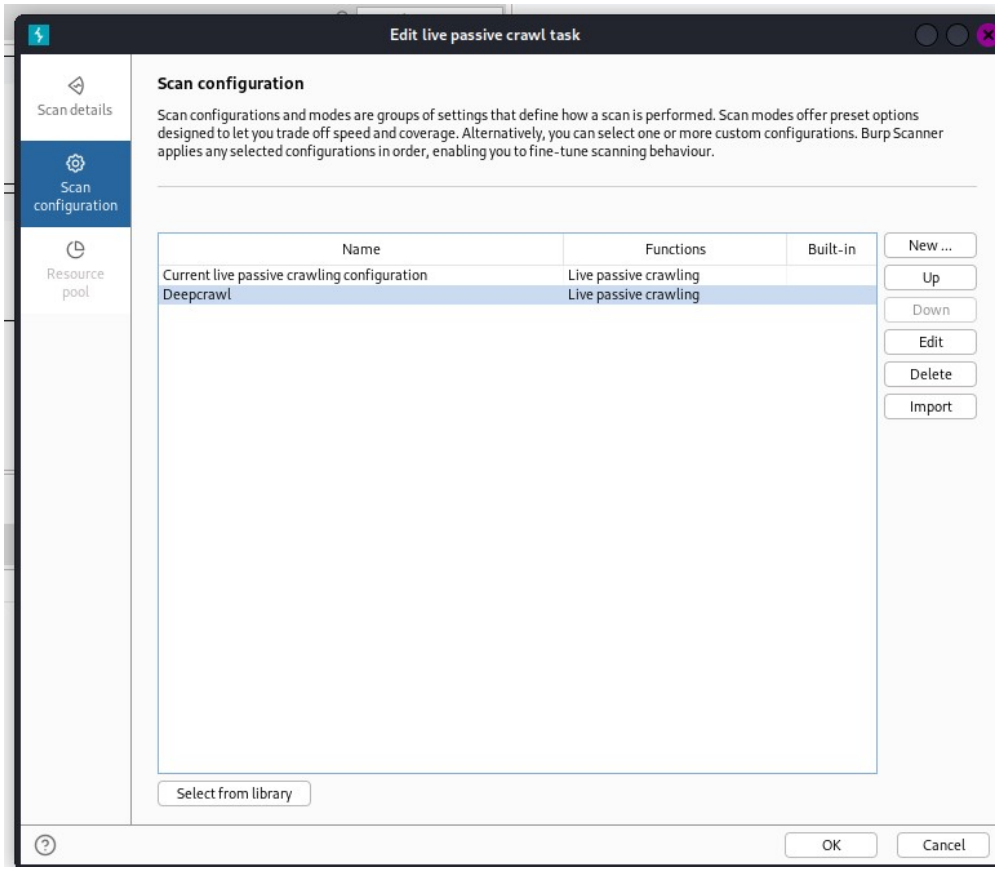
Then we can add the target website to our scope to continue tracking its traffic.



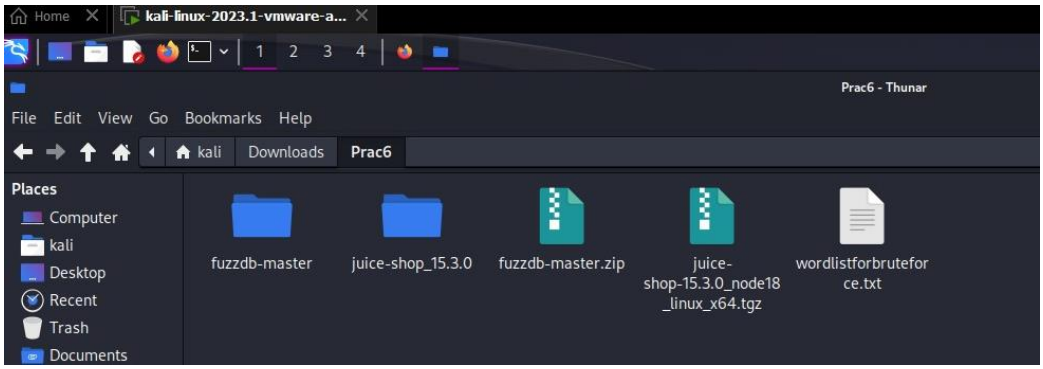
We can also create our own customized passive crawlers by using the following steps.



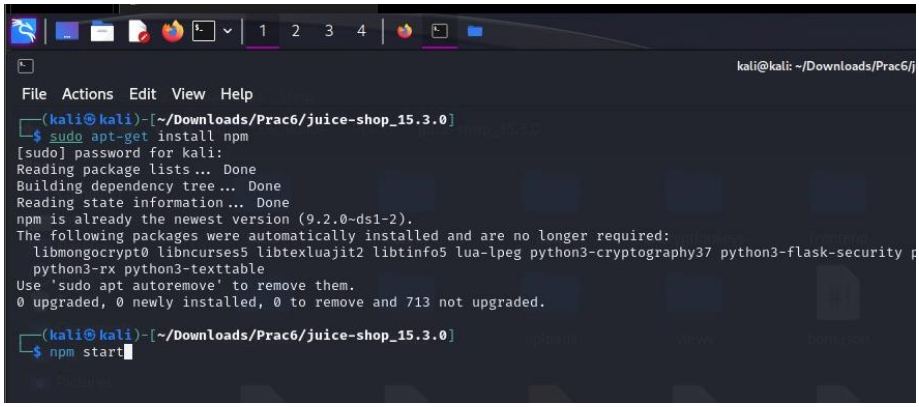




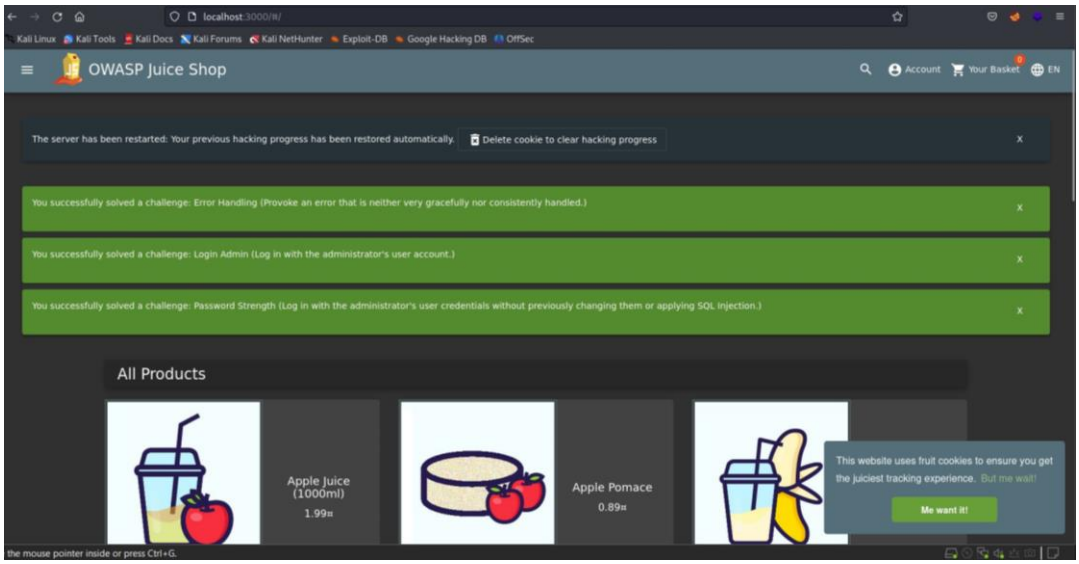
Then we will download fuzzdb master, juice shop and a wordlist and extract it.



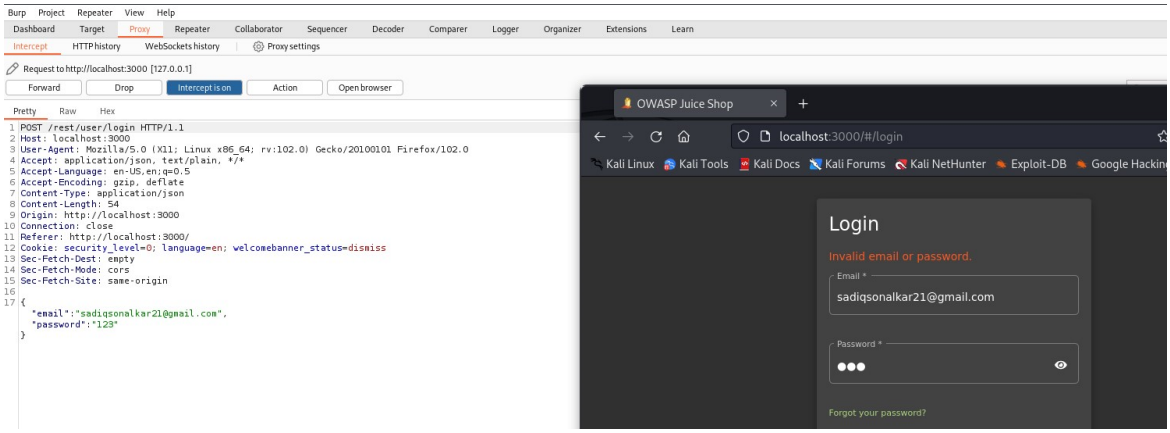
Then in the juice shop folder we will install npm and then start npm.



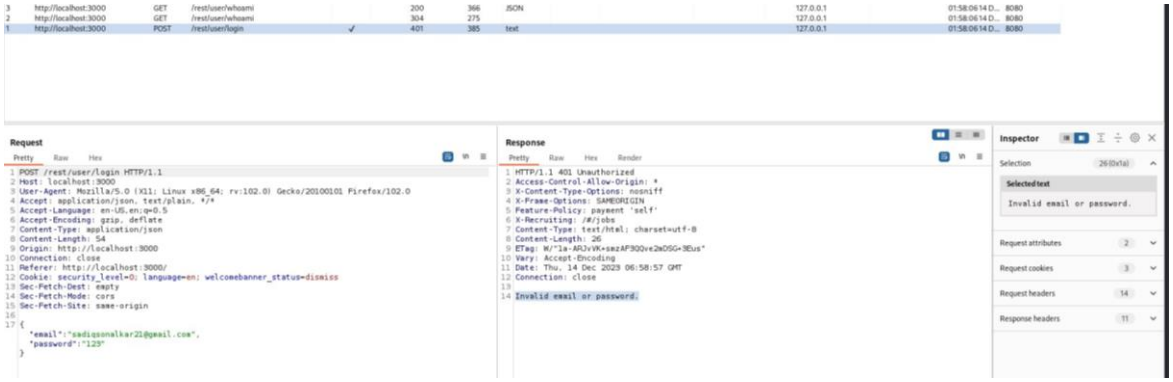
And it will start then go to a browser and enter ‘localhost:3000/’ and it will show our juice shop webpage.



Turn on the interceptor and try to login using any email id and password.



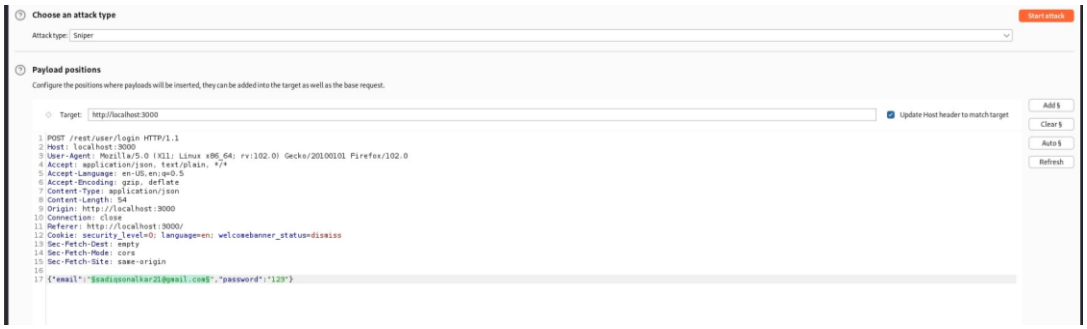
Then go to HTTP history and we can see a post request.



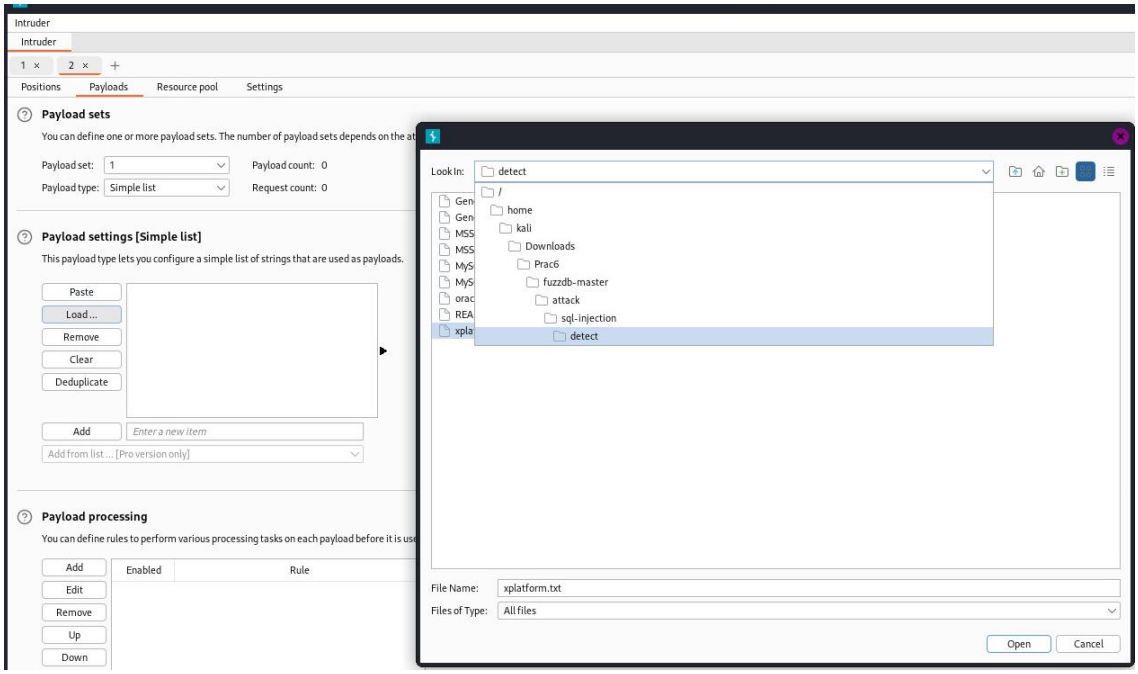
Copy the invalid email or password message.

Right click on the request and then select Send to Intruder.

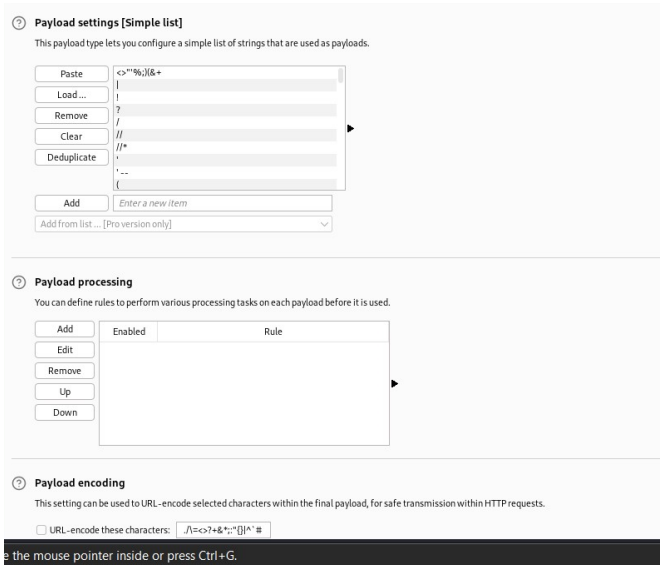
Highlight the parameter that you want to attack, in this case we will target the email to find what email is used for admin login.



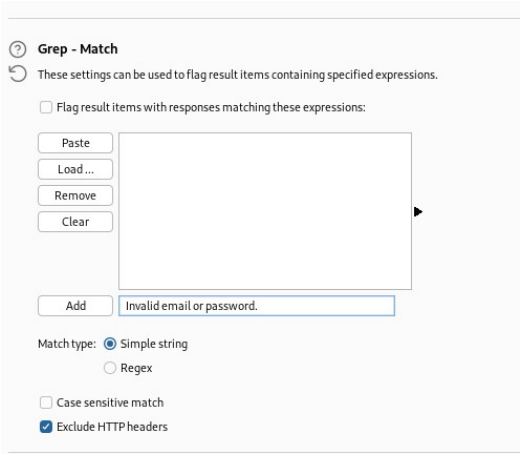
Select the payload tab, click on load and then navigate to the wordlist file provided or downloaded to you and select the appropriate option



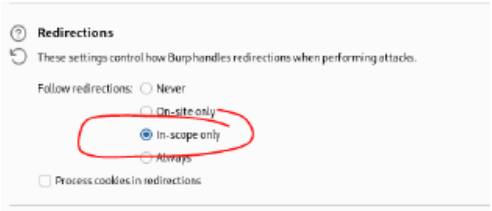
This should load the strings in the payload. Also uncheck the URL encode option below



Now select the settings tab and the clear the Grep-Match options and add the text that you had copied earlier. This will help us give a flag if any of our brute force doesn't work it will show the message.

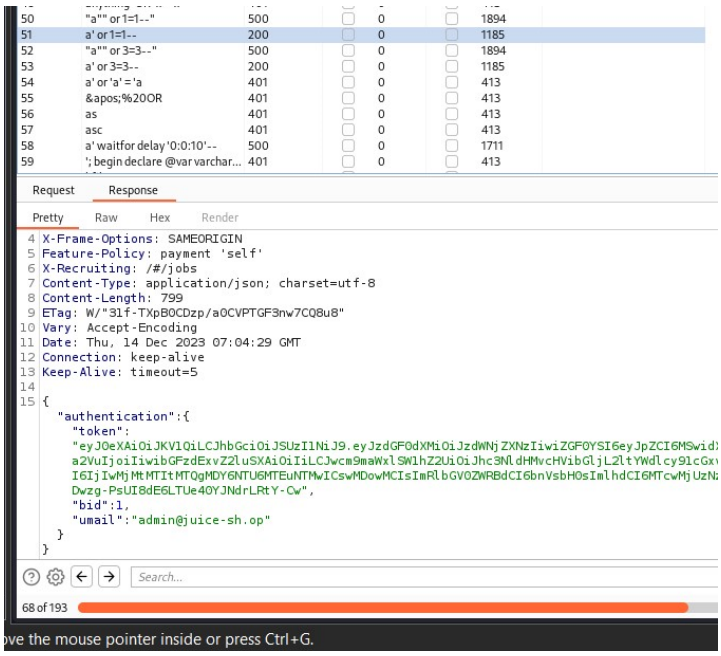


Scroll down a bit and select the In-scope option. This will ignore if there is any caching of pages on the website

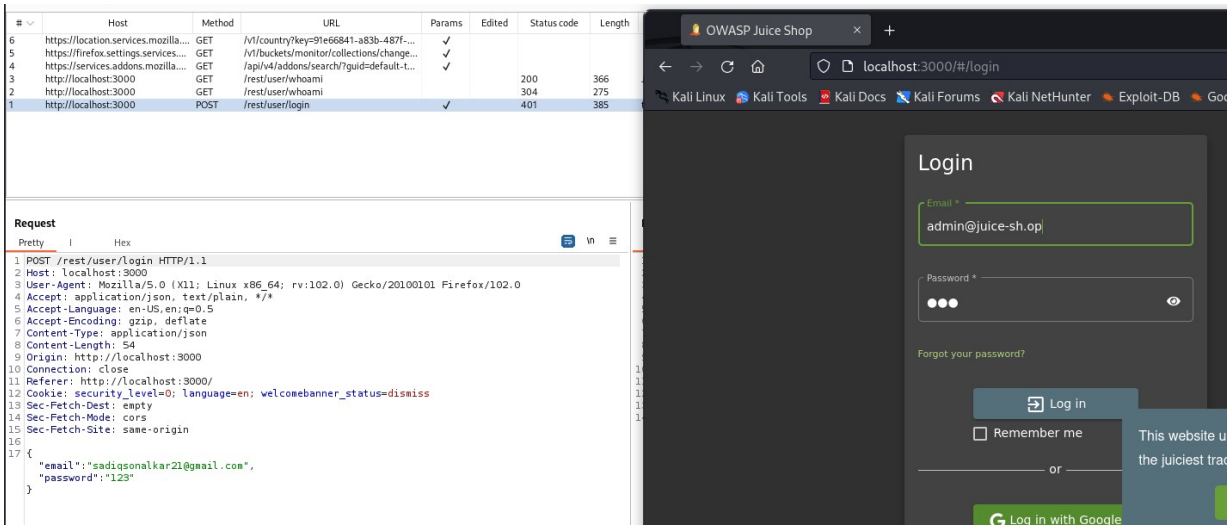


Now start the attack

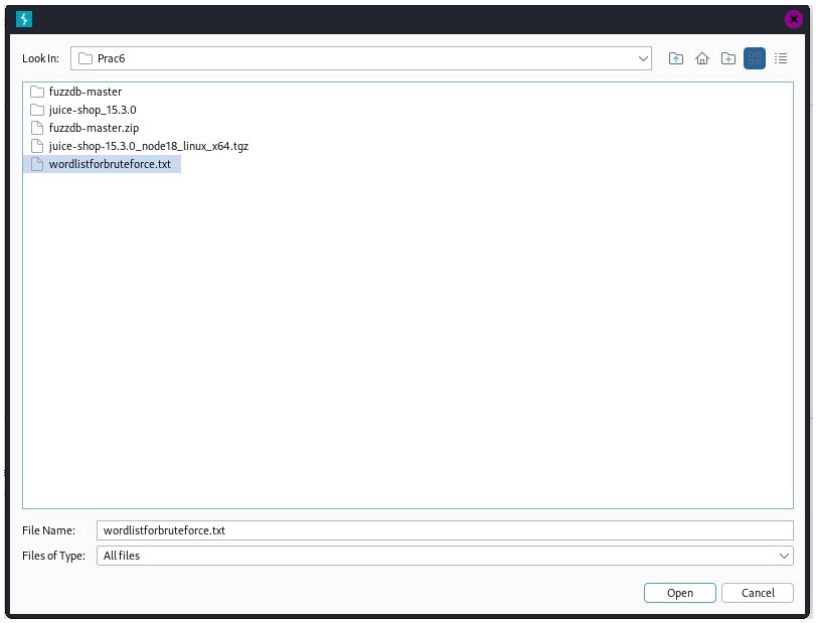
The attack starts. Now search for any request, which gives a 200 status code and token. Within that response you will notice the admin email too. Copy that admin email somewhere.



Now I know the admin email. But I don't know the password. We will need to brute force the login page to get the password. Close the attack session and let's use a wordlist payload to get the password.

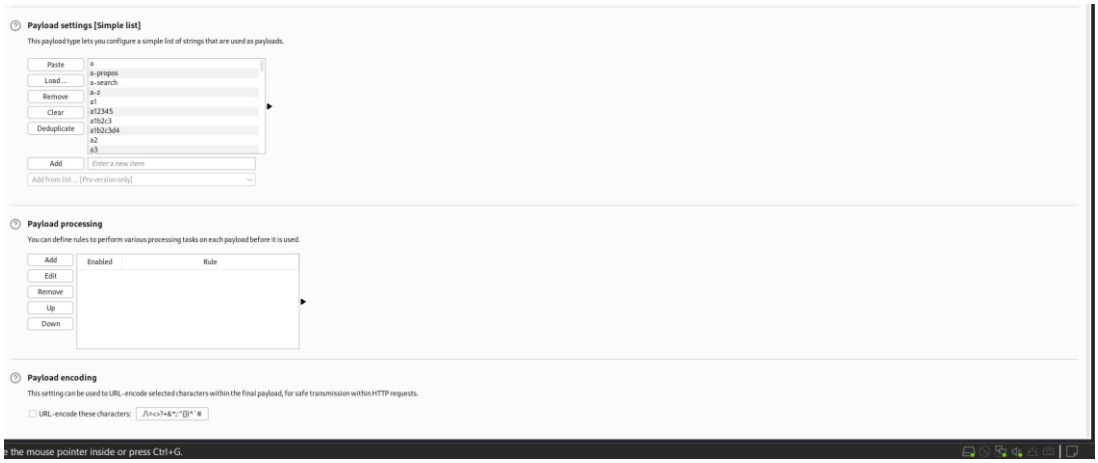


Clear the payload and load a new wordlist payload

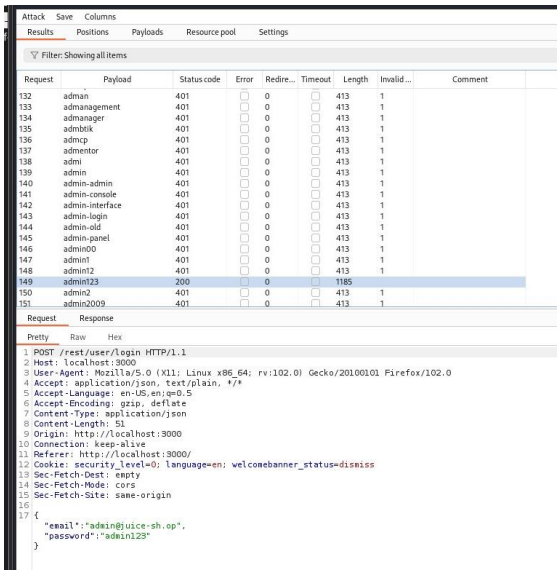


Disable the URL encode this character. Then add the inavalid email or password we copied before. Also select in-scope only.

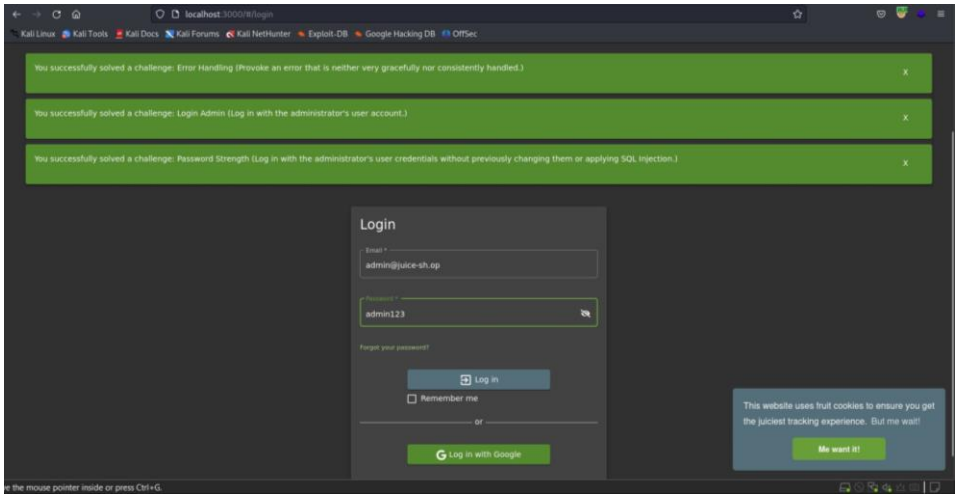
We are ready now to brute force the password field to get the login credentials. Start the attack.



You will notice a 200-response status code on a payload. That payload also does not show Invalid Email or Password error as 1. This means the payload text is the password of the login page.



Now we have got the email and password of the website. Let's try to login the website.



We are now able to access the admin control panel