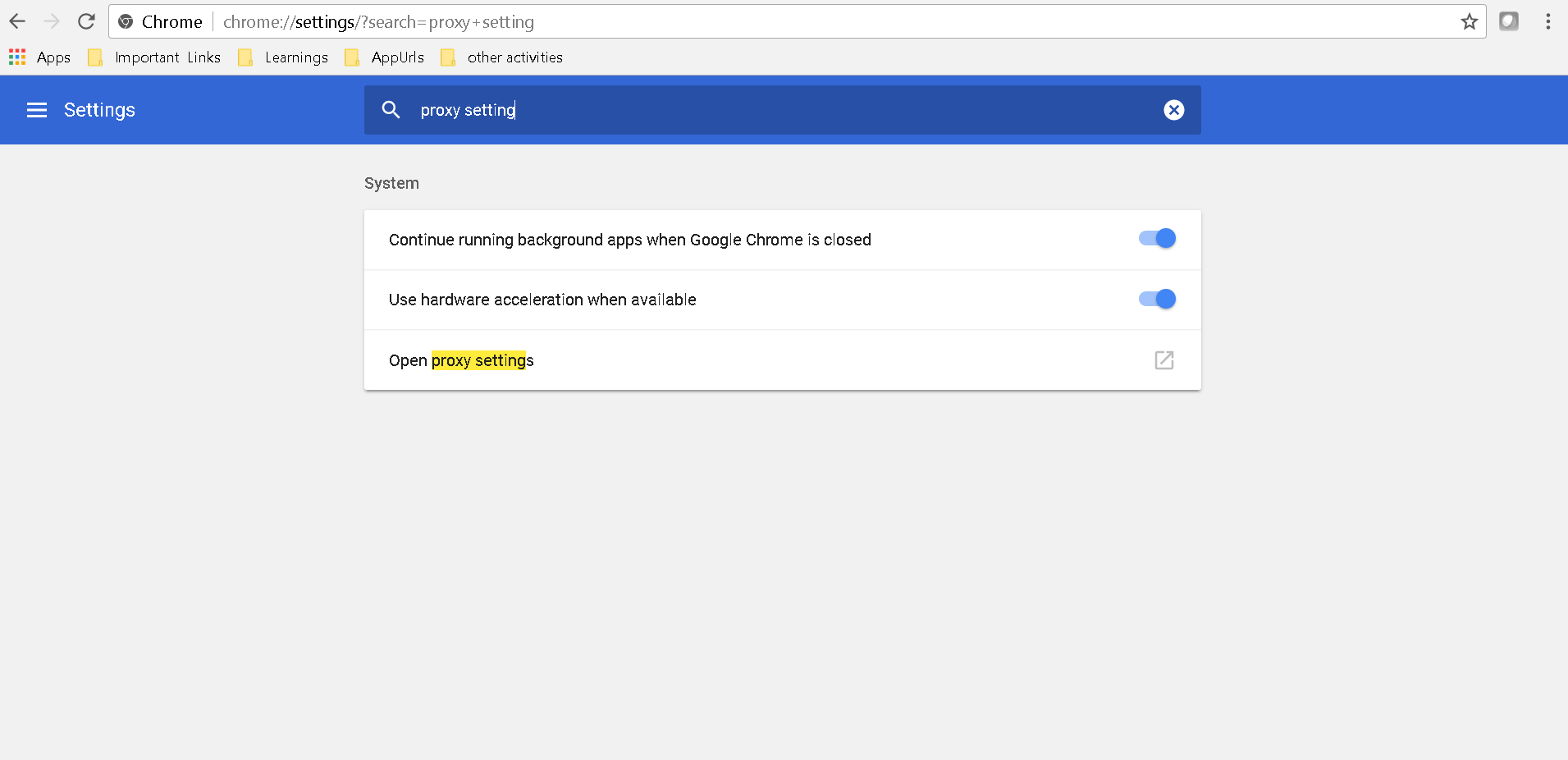
**Steps to Reproduce Hawtio Security Issues**

Requirement:

Burp Suite -download JAR file from <https://portswigger.net/burp/freedownload>.

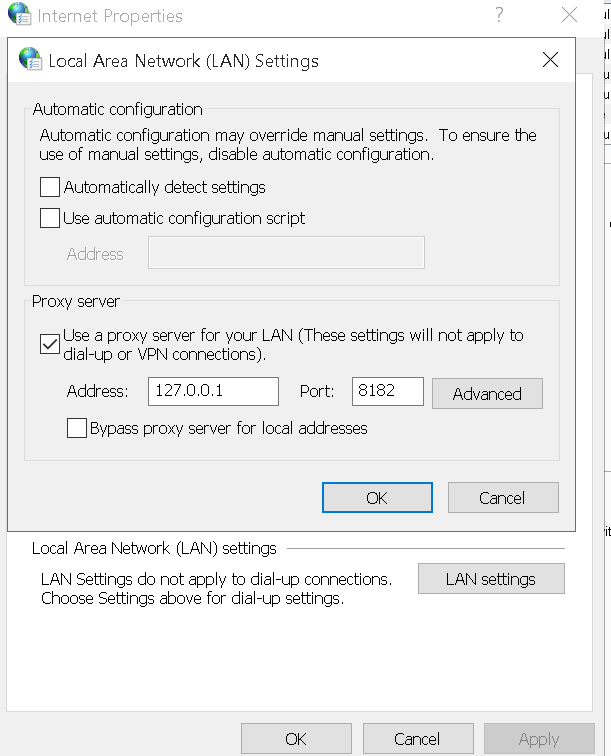
Steps:

Download Burp suite jar in local machine ,Go to Open proxy settings In the browser by searching for proxy settings in the browser settings

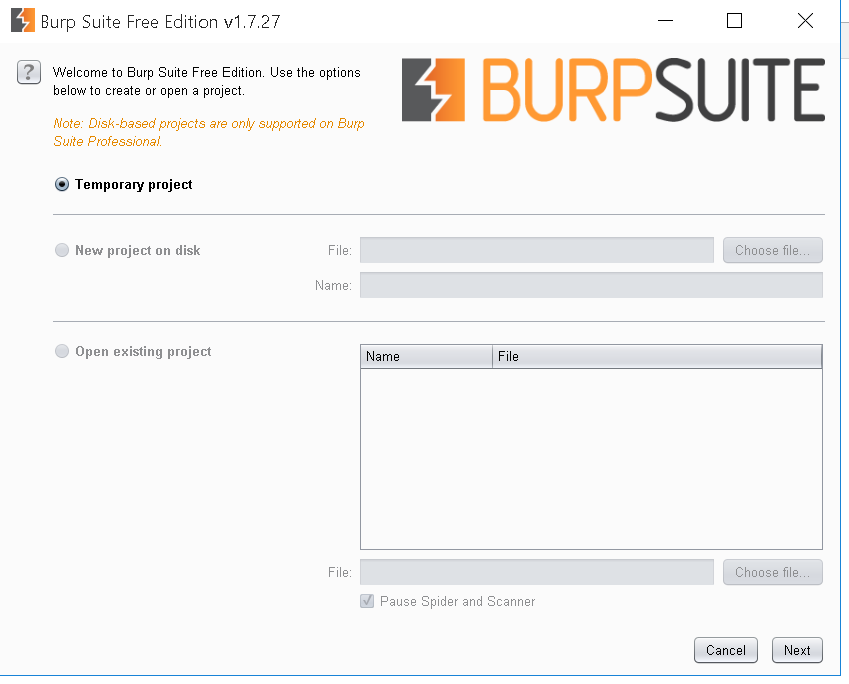


Click on the LAN settings in the pop up unselect “Automatically detect settings” and select “Use a proxy server for your LAN” under Proxy server.

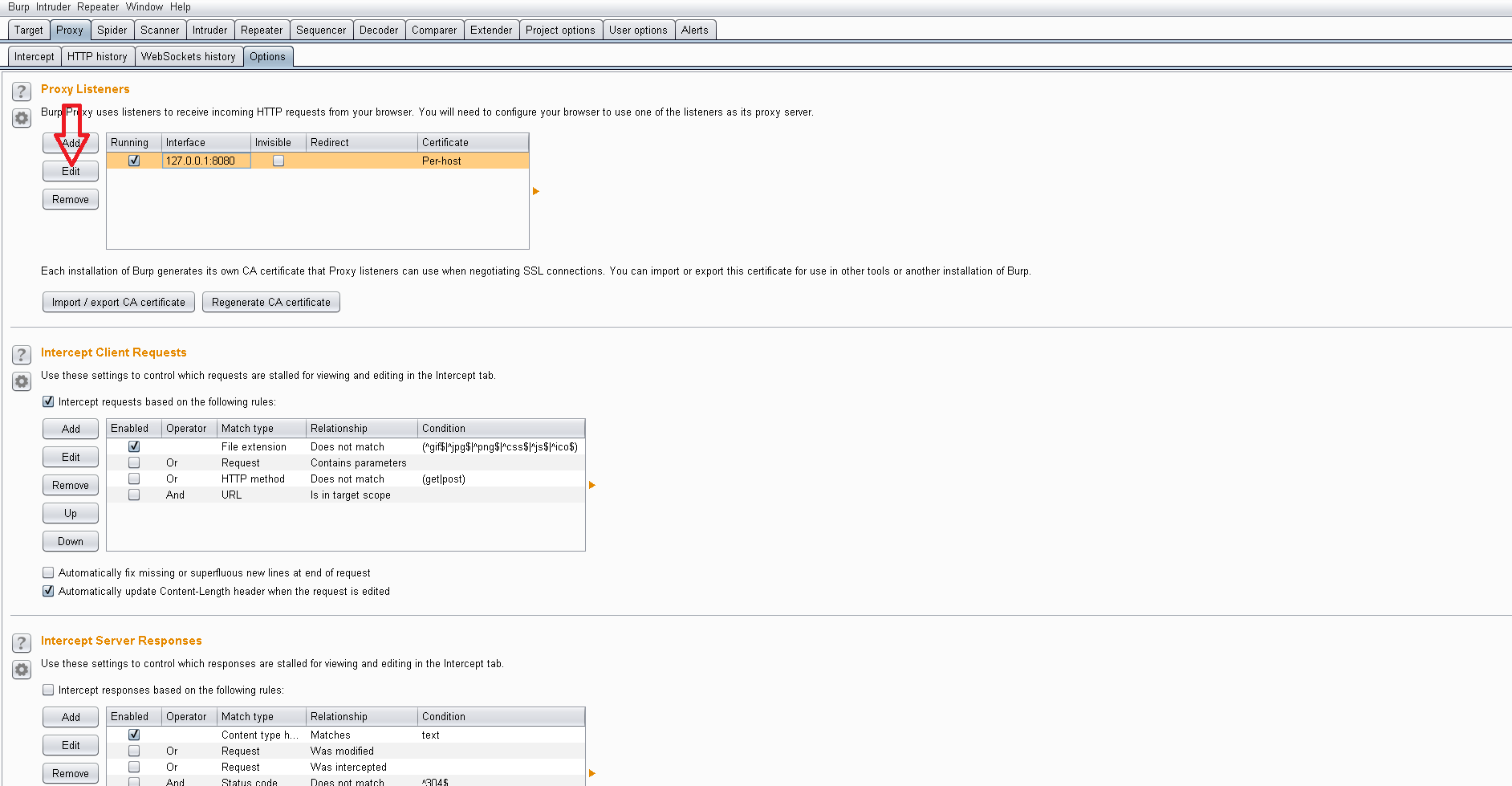
Give Address as 127.0.0.1 which is your local machine and give any port number which is not in use and then click ok .

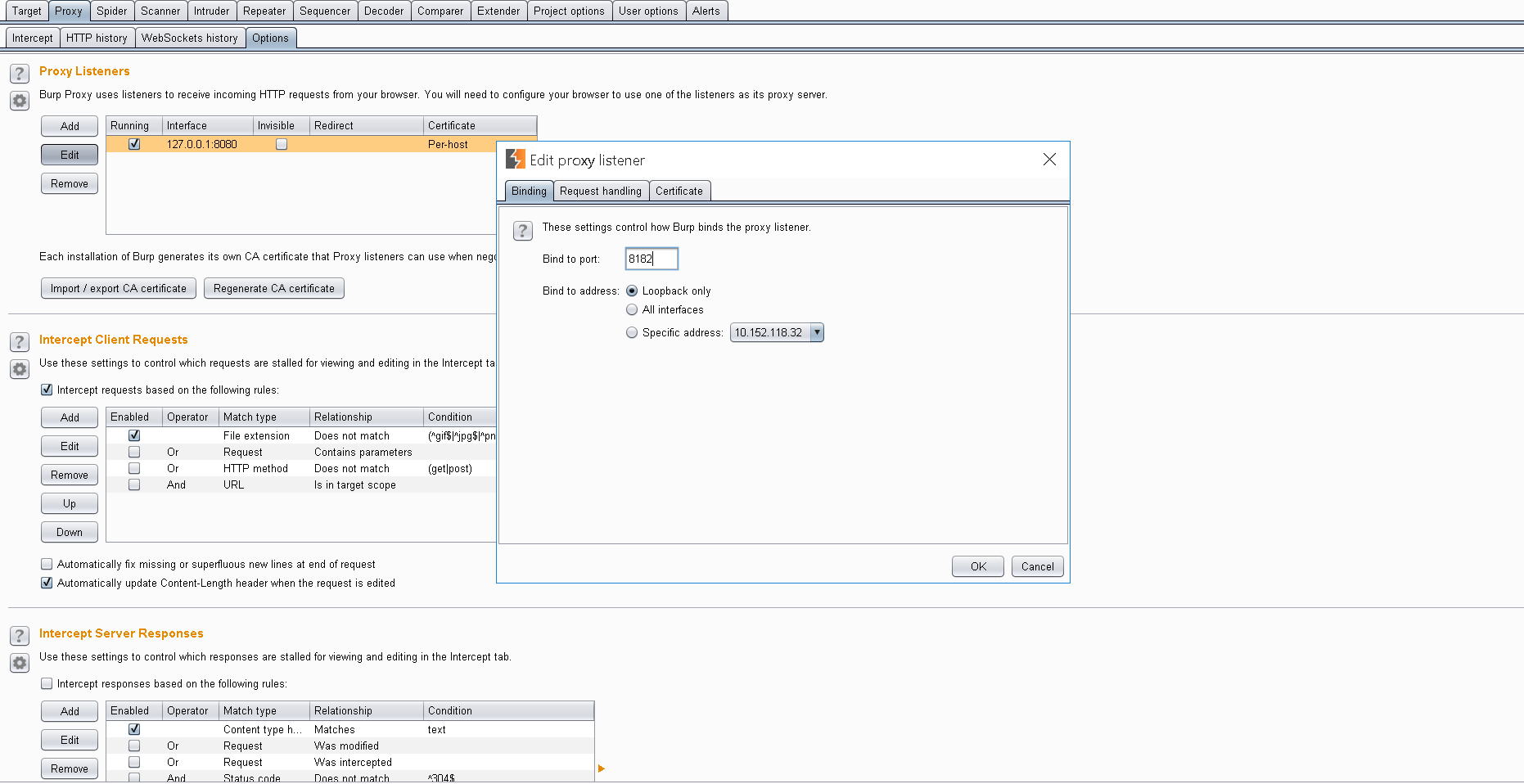


Now open Burp Jar and select Temporary Project and click next . Then click on Start Burp in the next screen. Go to proxy tab select options sub tab under proxy.

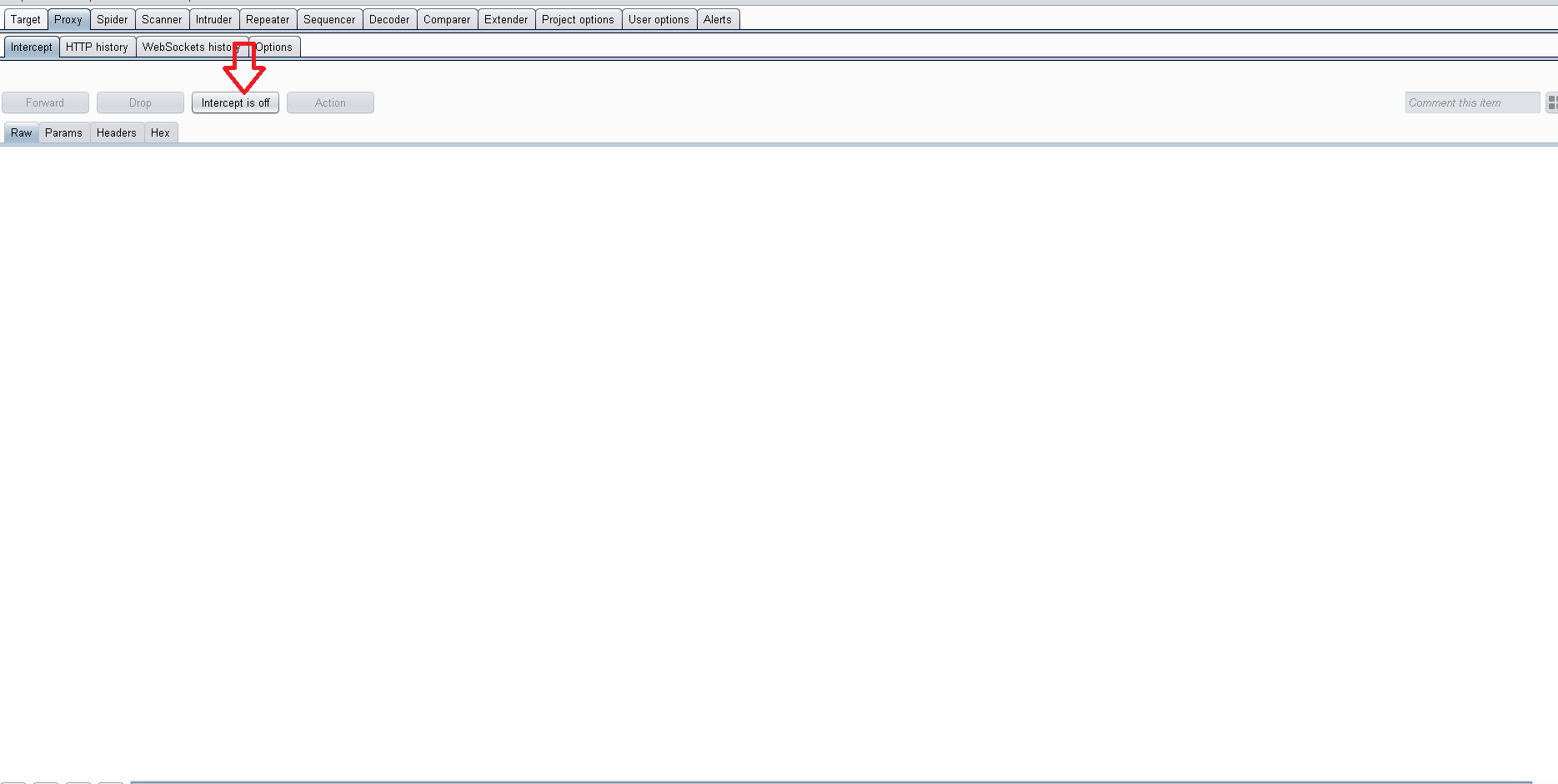


Under Proxy Listeners click on the row in the table and do edit .Change the port to the same port number which you have given in the browser proxy settings .

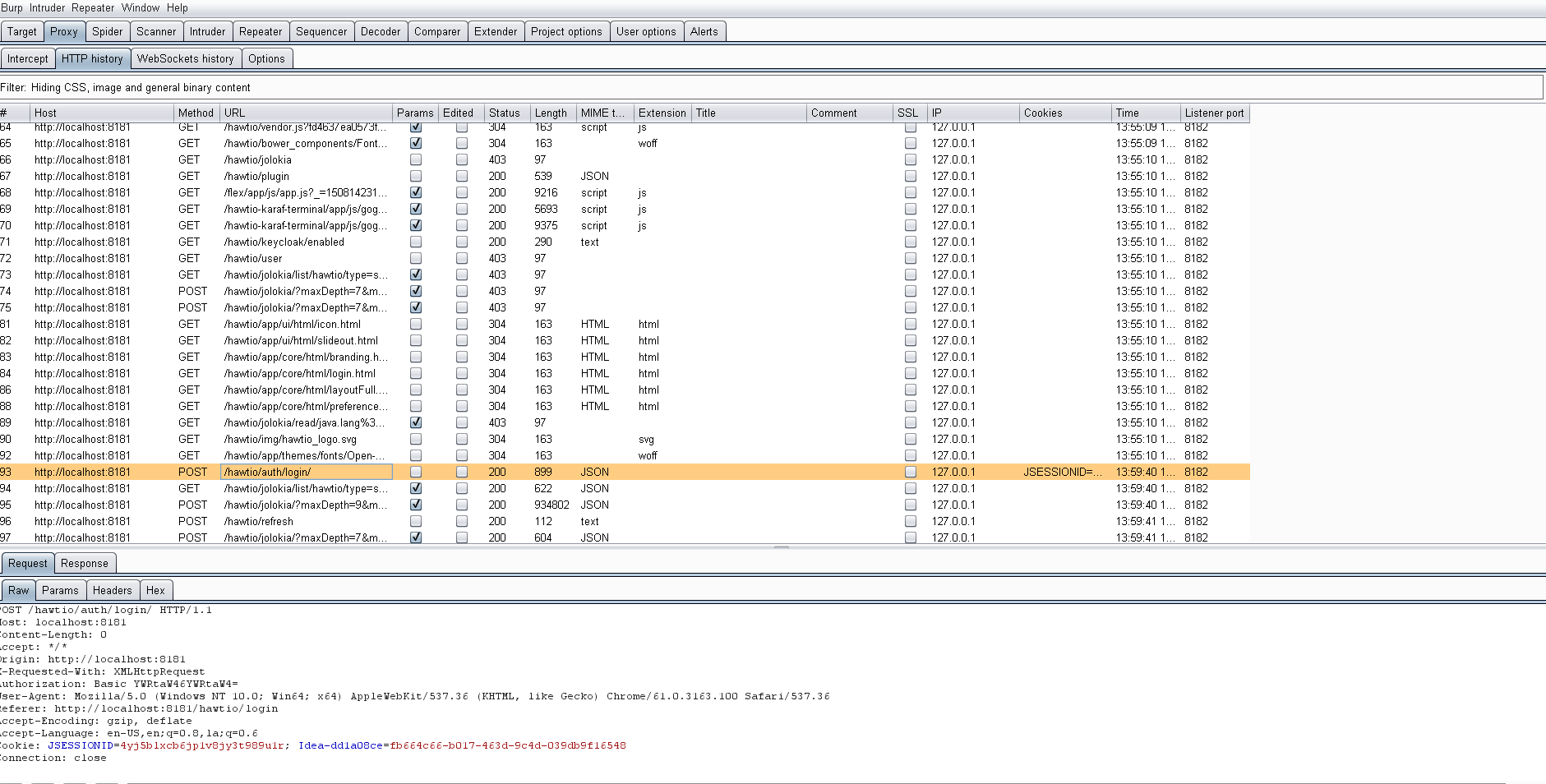




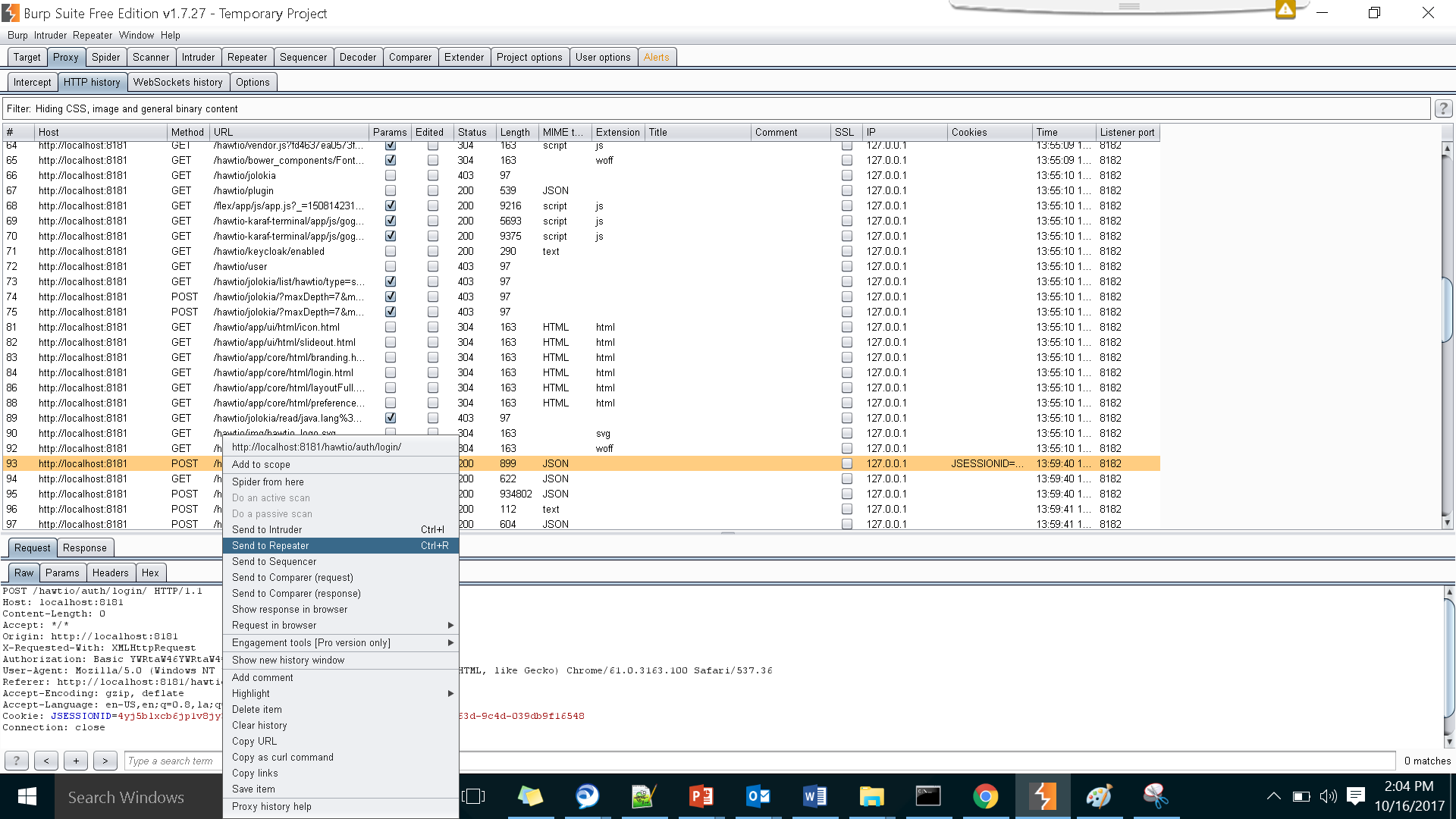
Now hit the Hawtio URL in the browser and go to intercept tab under proxy and turn off intercept .



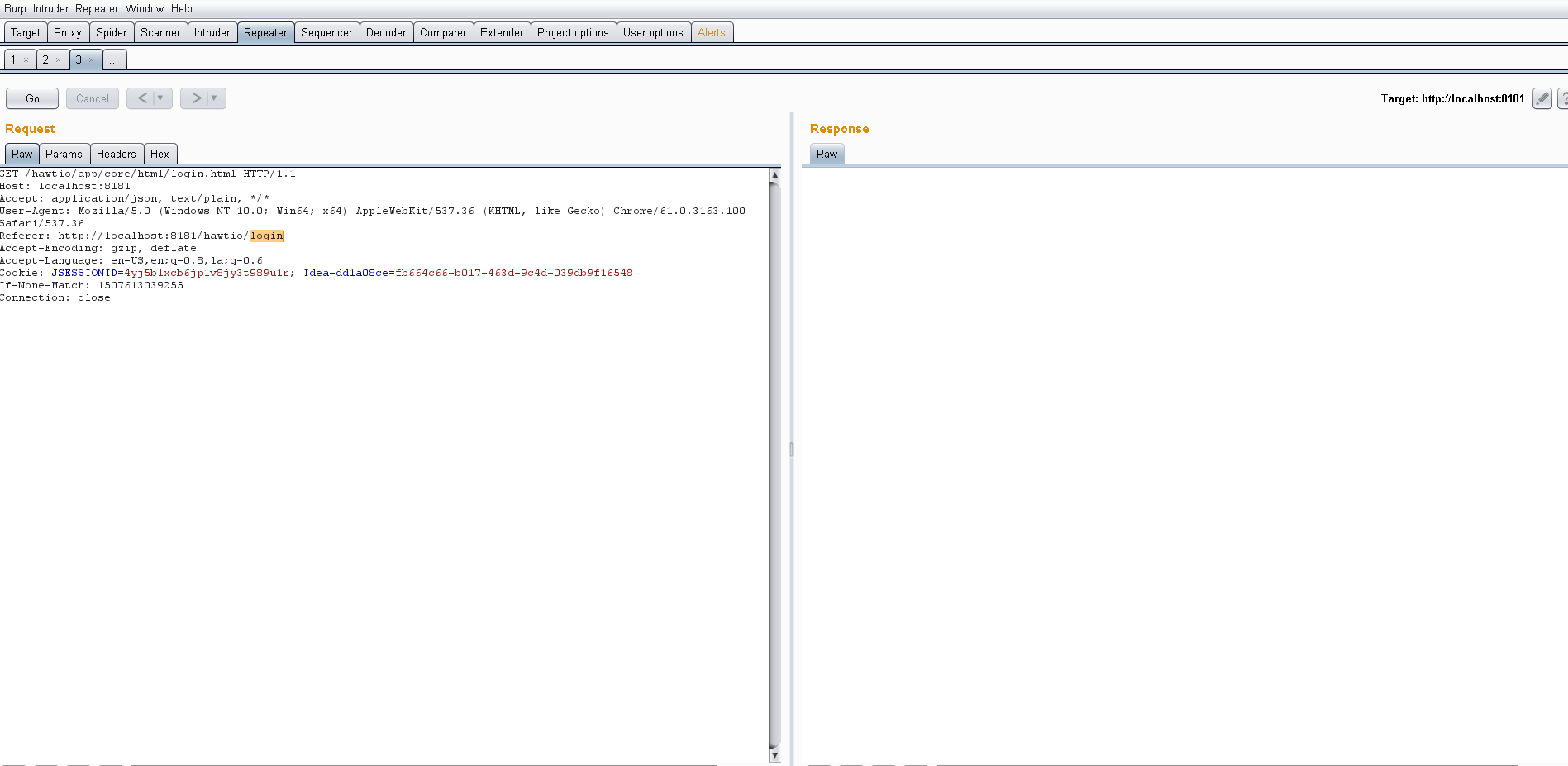
Under HTTP history you can find the request URL with response details. This is the same request sequence as the appscan report shows.



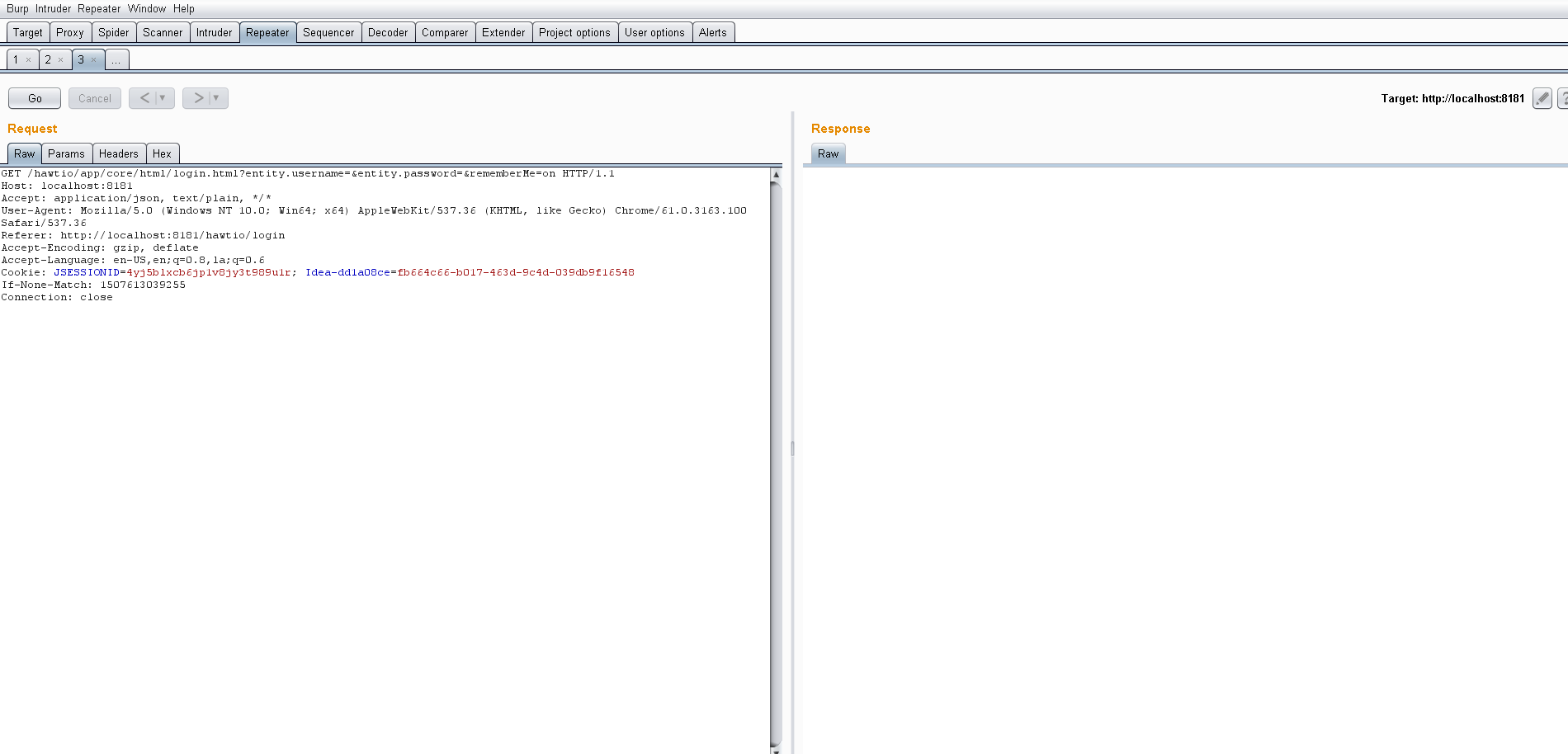
We can right click on any request and send it to repeater and repeat the request with modified param or HTTP verb.

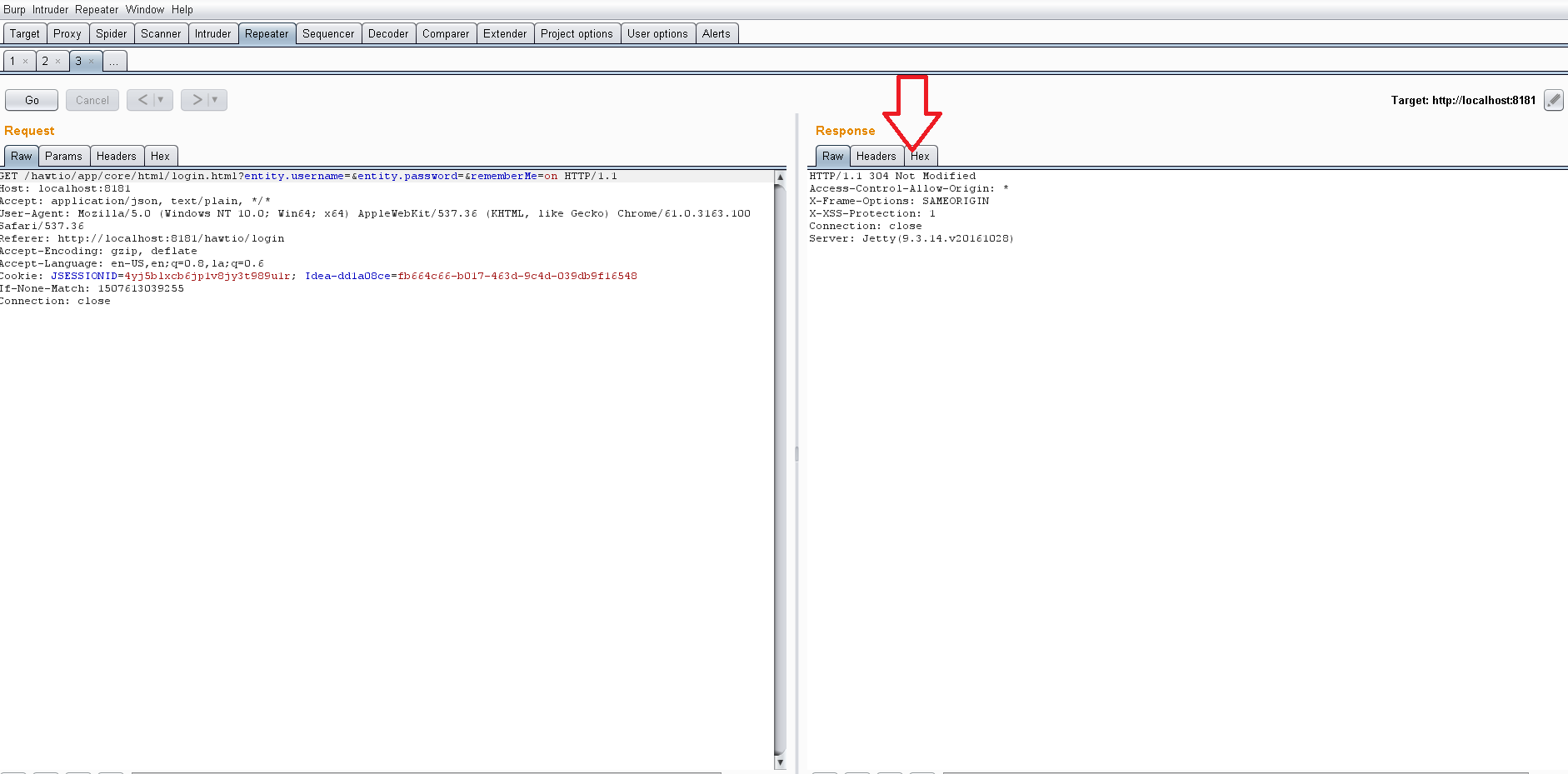


Under Repeater tab you can see the request which you have sent to repeat.

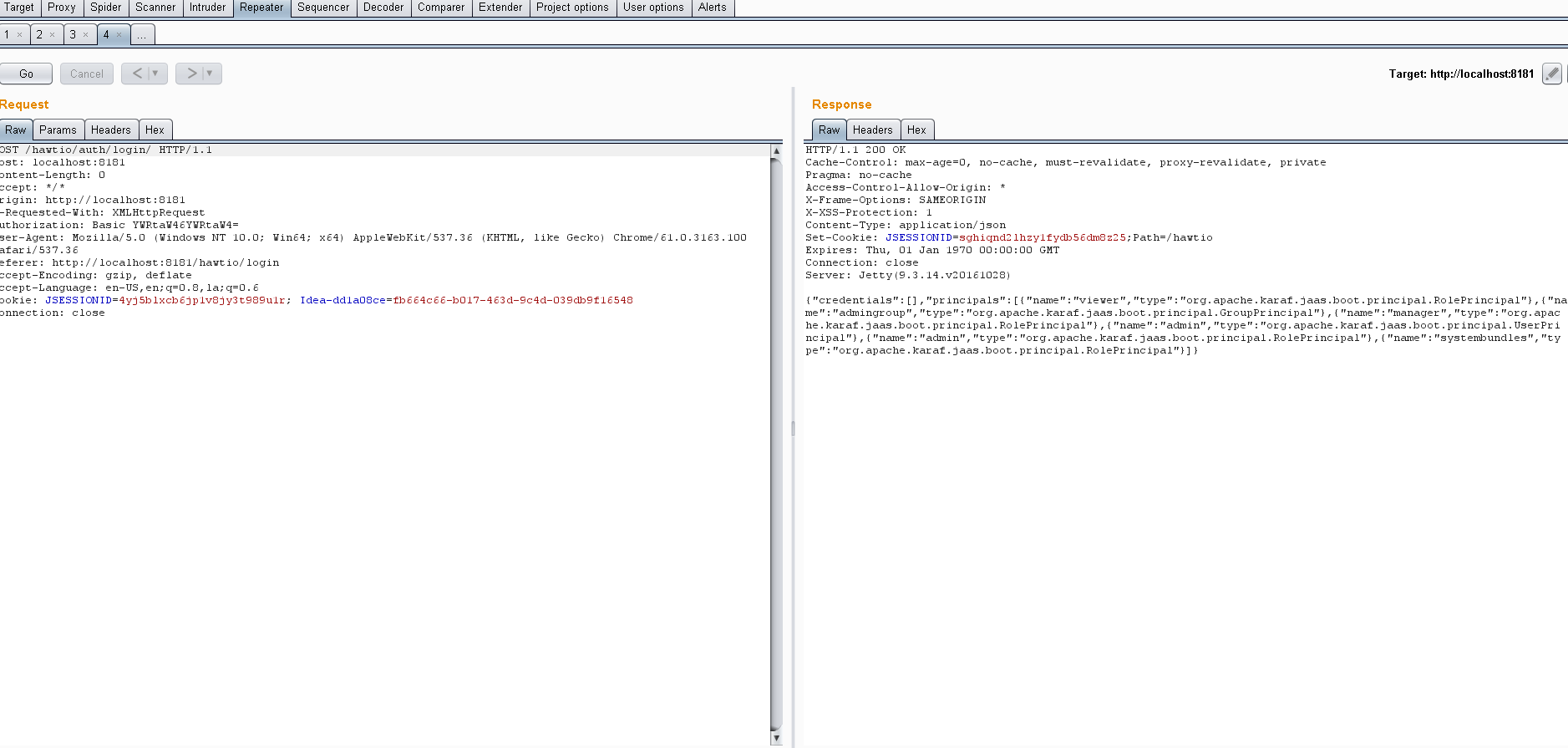


Below is the modified request which we can send using repeater by clicking on the Go button .

Username & password parameters are added to the url to check for the vulnerability.The response clearly shows whether it is vulnerability or not .There is no session id is created for this request, whereas the valid request creates the JSESSIONID by making server call. Hence the request Url is not valid or its not vulnerability



Response for the valid url.



Securing JDA Connect

REFS UI : JDA Connect UI is built using REFS framework. It uses basic karaf authentication. The Basic authentication protocol is a simple username/password authentication mechanism that is integrated into HTTP and is supported by most Web browsers.

The karaf authentication implemented is Basic Auth which is generally implemented by using a Filter which filters every request fired and checks the Authorization HTTP header of each request for the authentication key which contains a string which is in the following format “Basic <username:password>”.

The string <username:password> is base64 encoded so that all the characters in it are HTTP compatible. This string is intercepted by the ContainerRequestFilter and decoded to get the username and password which are compared against the user database in apache karaf and authenticated if the authentication is successful the request is accepted and the server sends the response containing the requested data. Else it returns a response with a *Unauthorised 401* as the *HTTP Status code* of the response with a response message stating the same

The authentication mechanism enabled in the restful service jda connect’s service framework is JAAS based Karaf authentication which uses Basic Auth to authenticate the session.

Following steps indicate how integration of REFS client woks with application server

1. When a Connection is requested by the web browser to connect to the application module, it is first sent to the REFS client which then requests the application server on behalf of the browser, in other words REFS server acts as a proxy.
2. Apart from acting as a proxy, REFS also stores some information like the authorization cookie in its memory and instead assigns the user or client its own session and session id.
3. When the request is made REFS attaches the authorization cookie along with the web Request before sending it to the server, which the server would use to authenticate and send appropriate response.