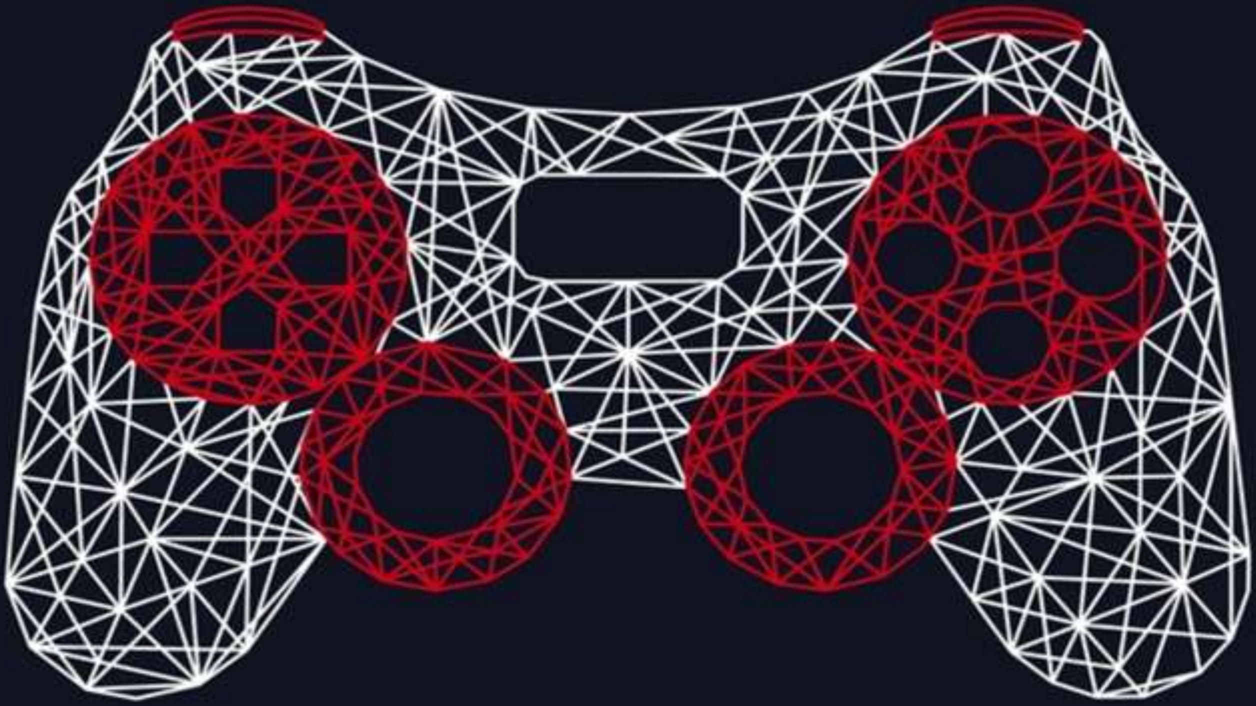
iGNITE Technologies

# BURP SUITE FOR PENTESTER



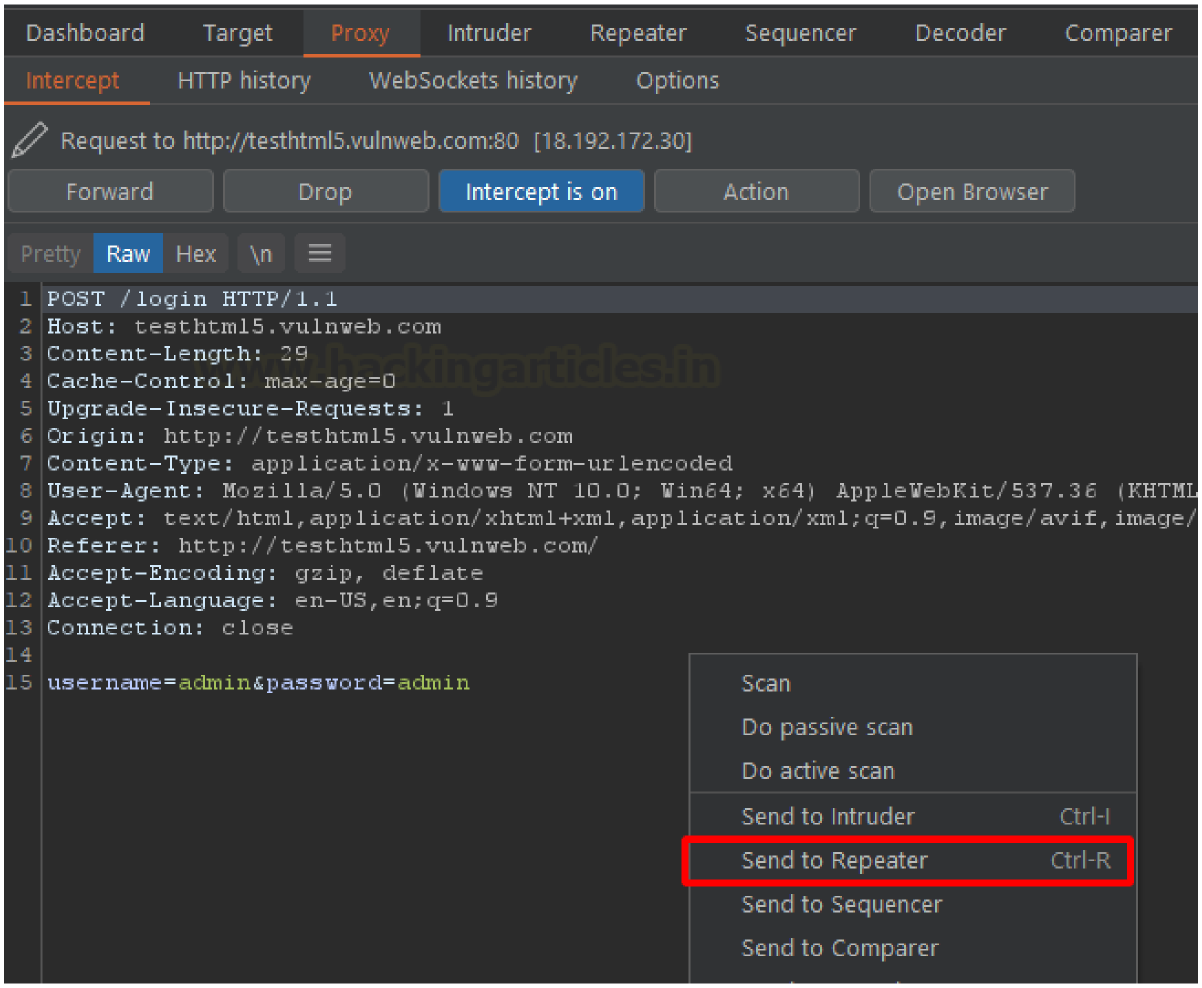
# REPEATER

WWW.HACKINGARTICLES.IN

## Contents

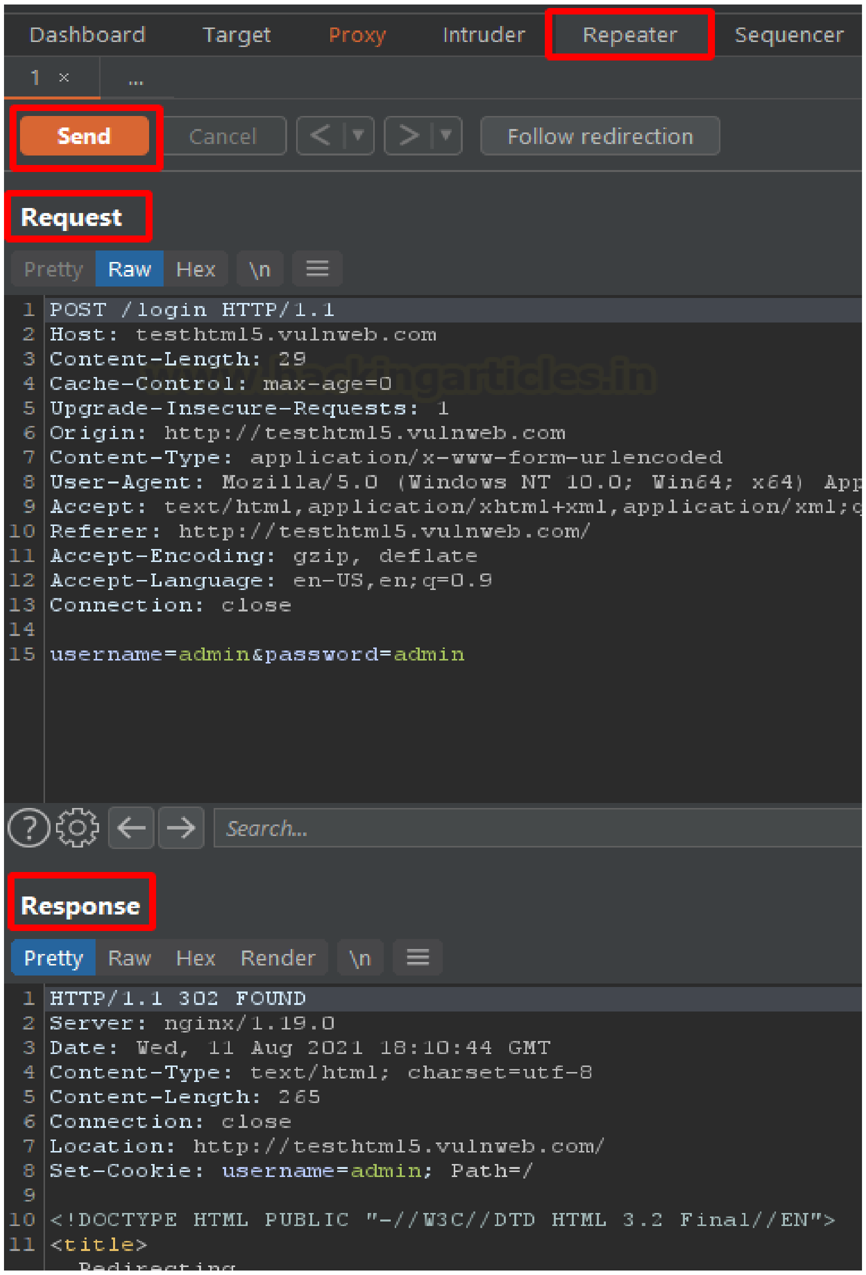
## Introduction

Even the users that are just beginning to use the Burp Suite are likely to be familiar with the functions of the Repeater. However, let's go over them to recall the functionality of the Repeater. It is a tool that is designed to allow the user or attacker to change or resend particular HTTP requests and understand and analyze the response generated by it. Here in the demonstration below, we have captured an HTTP request from the browser. This request is captured inside the Intercept sub-tab inside the Proxy tab. By Right Clicking on the captured request, a drop-down menu appears. This menu contains an option that says "Send to Repeater". Clicking this option will send the request to the repeater. You can also use the shortcut Ctrl and R to send any request to the repeater.



```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Content-Length: 29
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testhtml5.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
Referer: http://testhtml5.vulnweb.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=admin&password=admin
```

Now that we have successfully sent the request to the Repeater, we can move to the Repeater and toggle with the request, and observe the response generated for that particular request. In line with the Proxy Tab, we have the Repeater tab as well. Here, we can see that we have the request that was captured earlier. The request can be edited as per the user's requirements. Click the Send Button, the request will be sent to the target and the response that was generated will be presented in the Response section as shown below.

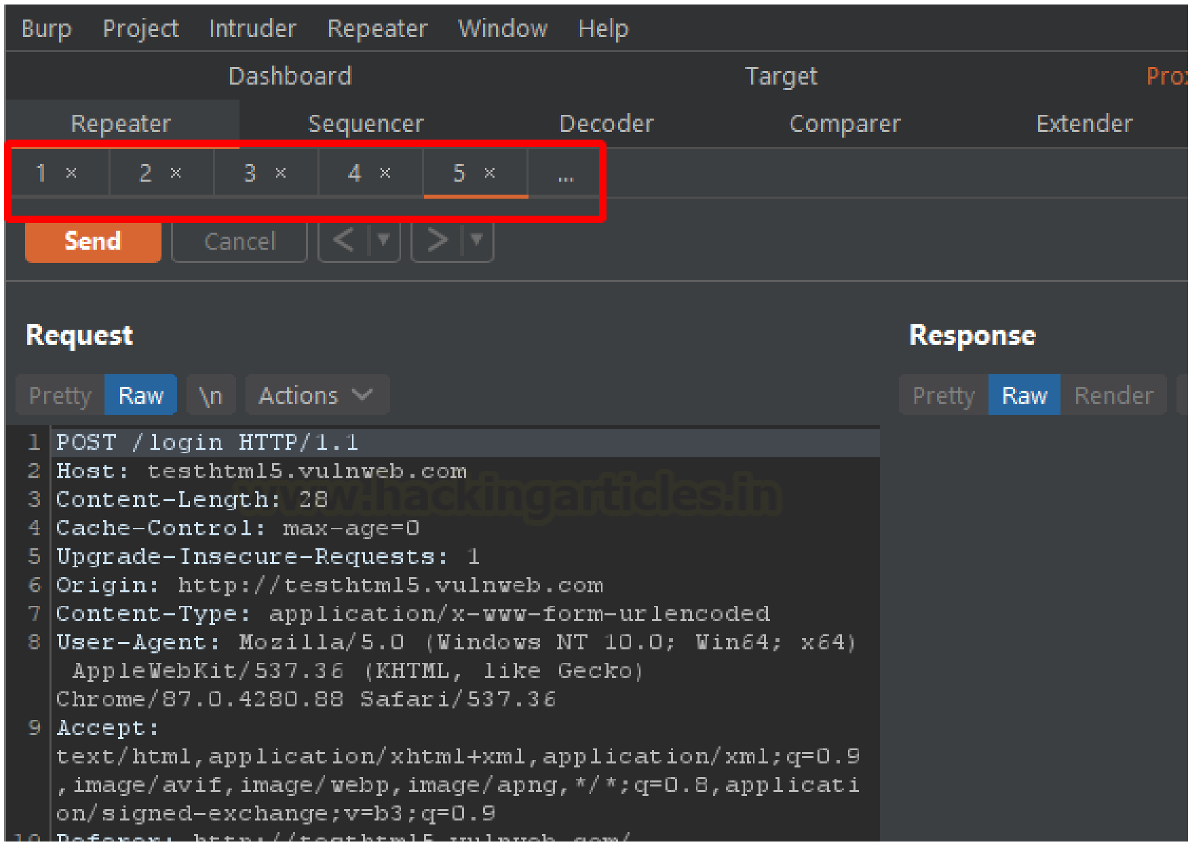


```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Content-Length: 29
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testhtml5.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) App
Accept: text/html,application/xhtml+xml,application/xml;q
Referer: http://testhtml5.vulnweb.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=admin&password=admin
```

```
HTTP/1.1 302 FOUND
Server: nginx/1.19.0
Date: Wed, 11 Aug 2021 18:10:44 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 265
Connection: close
Location: http://testhtml5.vulnweb.com/
Set-Cookie: username=admin; Path=/

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>
  Redirecting
```

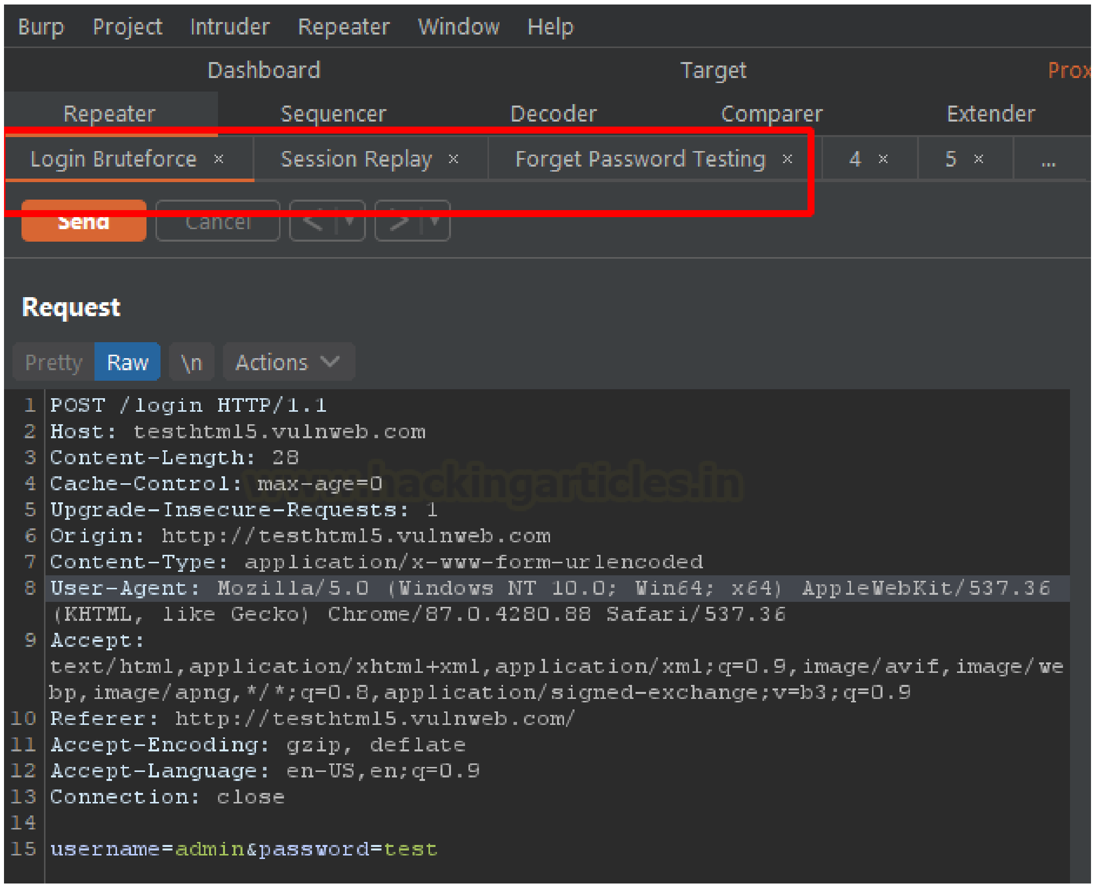
## Renaming the Tabs

During any assessment, the Repeater tends to get cluttered with multiple tabs with numbers as shown in the image below. Since each time, a request is being sent to the Repeater, the tab gets incremented by one. This can get confusing to catch up after a while of working on a project.



```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Content-Length: 28
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testhtml5.vulnweb.com
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/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
```

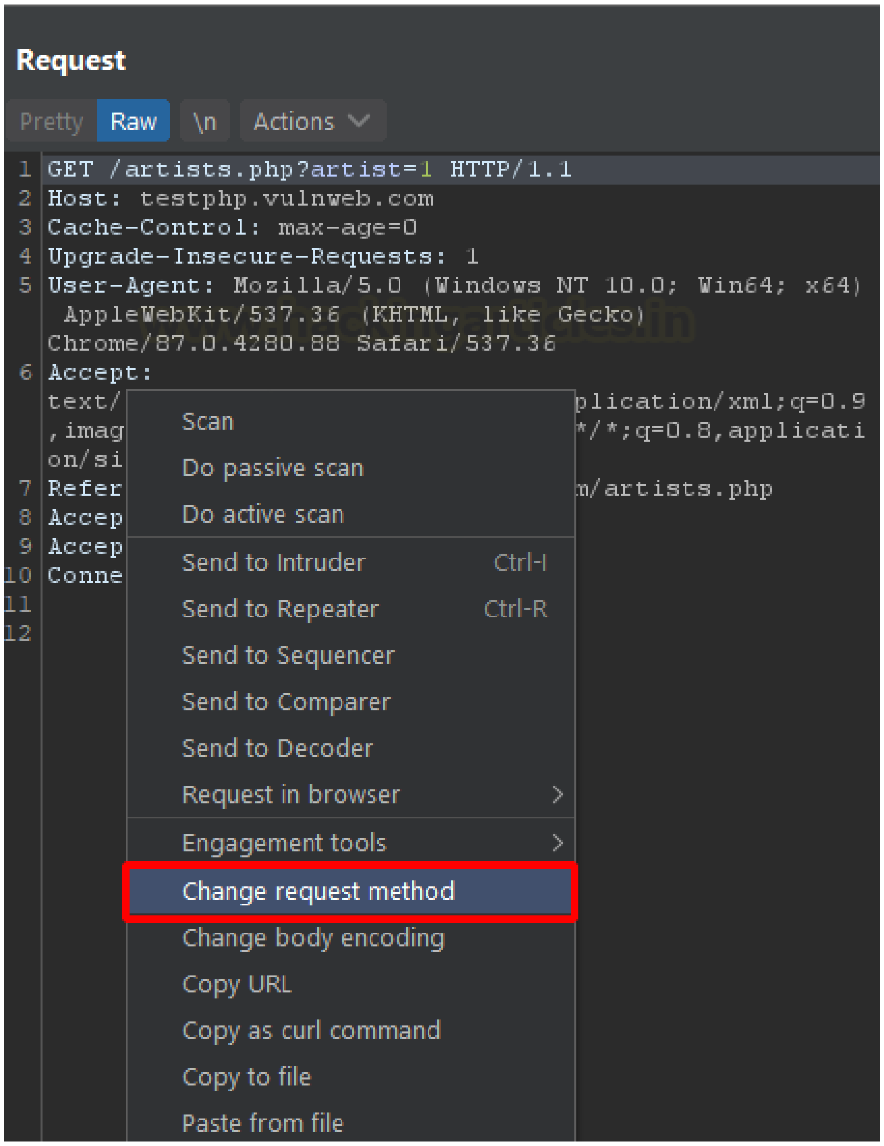
Upon double Clicking the Tab, you can rename the Tab based on your preference. This can be anything that can help you remember and sort. In the Demonstration below, we renamed the Tabs concerning the Testing that we were performing on that particular session.



```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Content-Length: 28
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testhtml5.vulnweb.com
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/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://testhtml5.vulnweb.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=admin&password=test
```

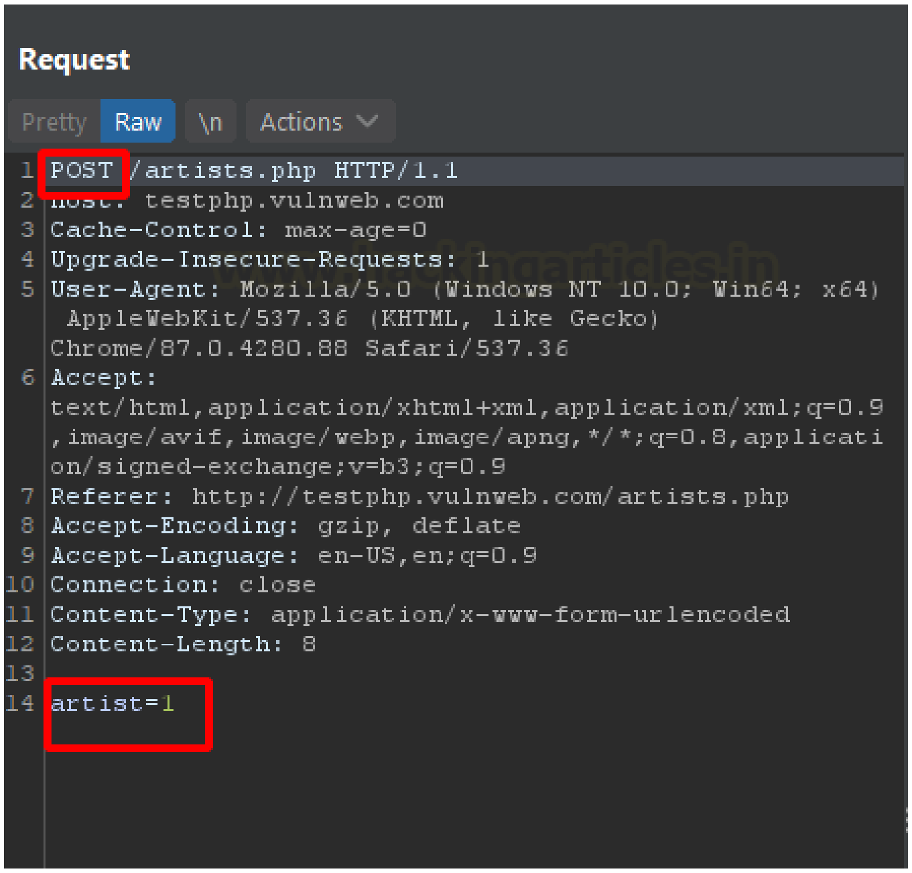
## Request Method

Next, we will discuss the ability of the Repeater to change the Request Method type. These are the HTTP methods such as the GET, POST, PUT, OPTIONS, etc. Some web pages are configured to work with multiple methods. Suppose, you want to request a page with input fields, to request that form, we need to use the GET methods, and to submit the data from the fields to the server, you will require the POST methods. This can be managed with just an option present in the right-click menu by the name of "Change request method". In the demonstration provided below, we have a GET request.



```
GET /artists.php?artist=1 HTTP/1.1
Host: testphp.vulnweb.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/                                        plication/xml;q=0.9
,imag                                        */*;q=0.8,applicati
on/si
Refer                                        m/artists.php
Accep
Accep
Conne
```

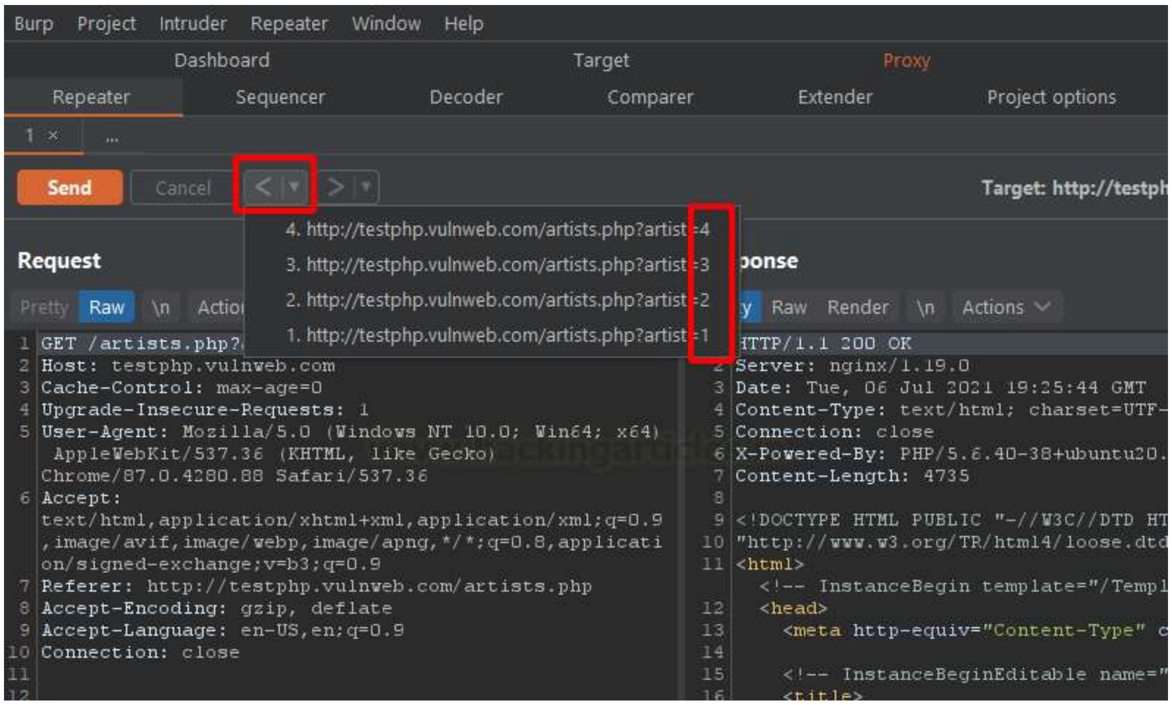
Upon changing the request method we can see that the method was changed to POST from GET and any parameters that were present in the URL were shifted to the body of the request as per the norms of the POST request.



```
POST /artists.php HTTP/1.1
Host: testphp.vulnweb.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/artists.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 8

artist=1
```

## Request History

When using any Web Browser, we have the Forward and Backwards buttons on the browser. These help us to navigate to the Previous or Next Page. The repeater has the Back (<) and Forward (>) buttons as well. These can help where we get a 301 Response for our request. This means that we can follow the Redirection. Using the Forward (>) button we can choose to move to the next request and its subsequent response. However, in case you are using the repeater to test different parameters and their effects on the response, then you can go back to any response that might be working as per your requirement. The Request history can be viewed by clicking the Drop-Down button next to the Backward (<) button as demonstrated below.

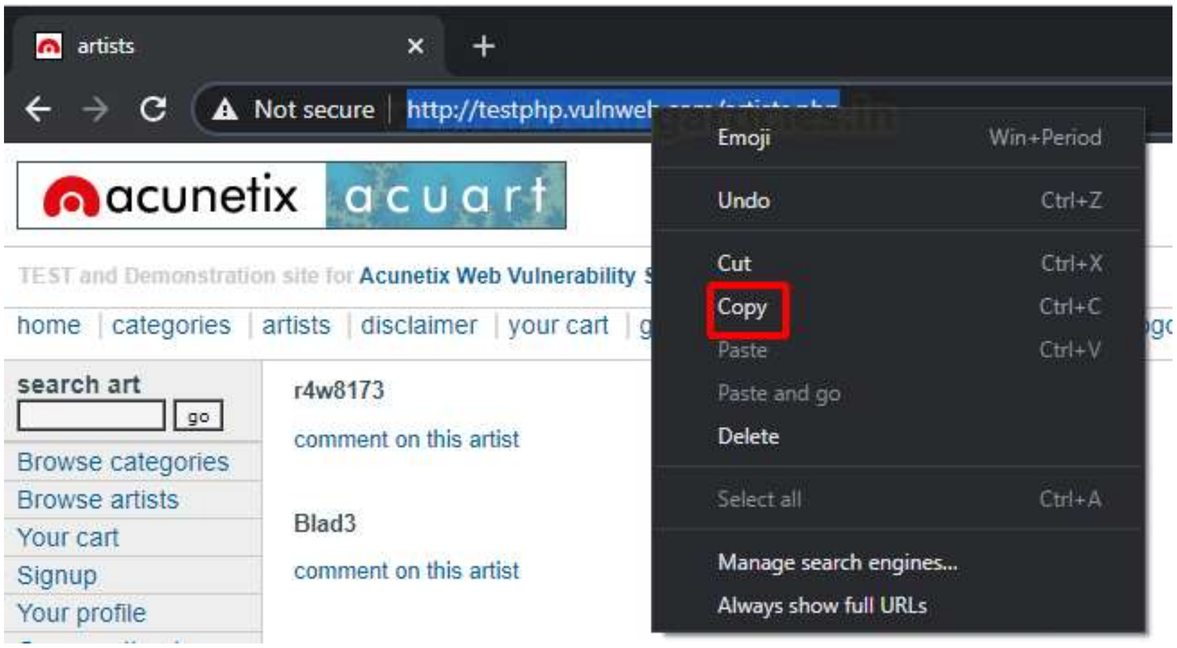

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy
Repeater Sequencer Decoder Comparer Extender Project options
Send Cancel
Target: http://testph
4. http://testphp.vulnweb.com/artists.php?artist=4
3. http://testphp.vulnweb.com/artists.php?artist=3
2. http://testphp.vulnweb.com/artists.php?artist=2
1. http://testphp.vulnweb.com/artists.php?artist=1
Request
Response
Pretty Raw \n Actions
Raw Render \n Actions

```
GET /artists.php?a
Host: testphp.vulnweb.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/artists.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

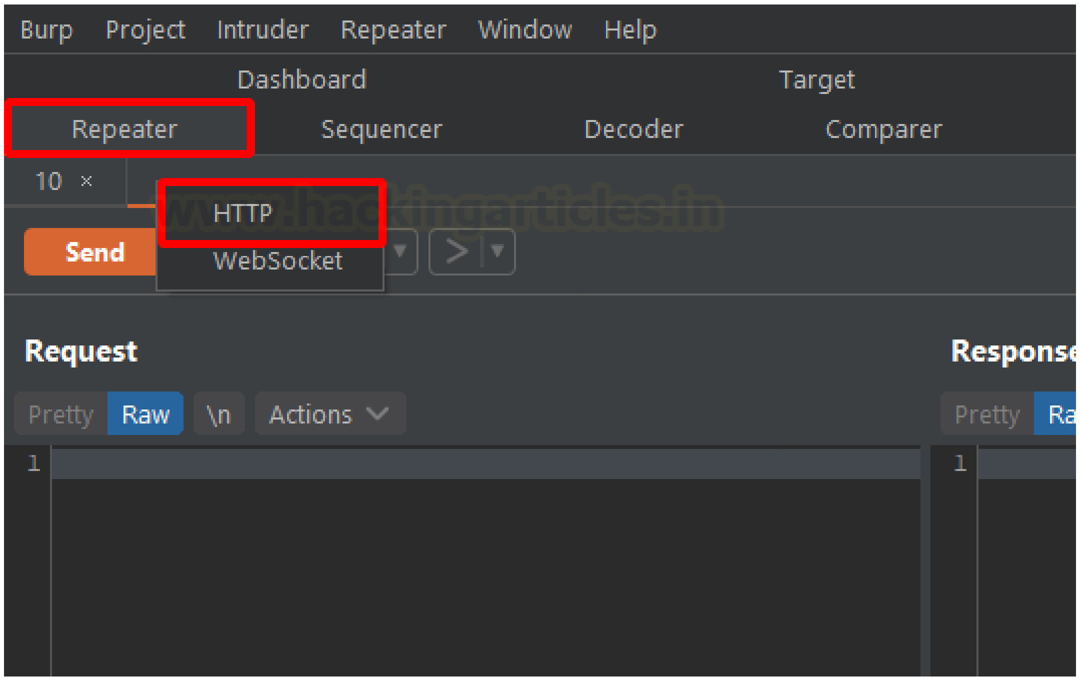
```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Tue, 06 Jul 2021 19:25:44 GMT
Content-Type: text/html; charset=UTF-
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.
Content-Length: 4735

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HT
"http://www.w3.org/TR/html4/loose.dtd
<html>
  <!-- InstanceBegin template="/Templ
  <head>
    <meta http-equiv="Content-Type" c

    <!-- InstanceBeginEditable name="
    <title>
```

## URL as Request

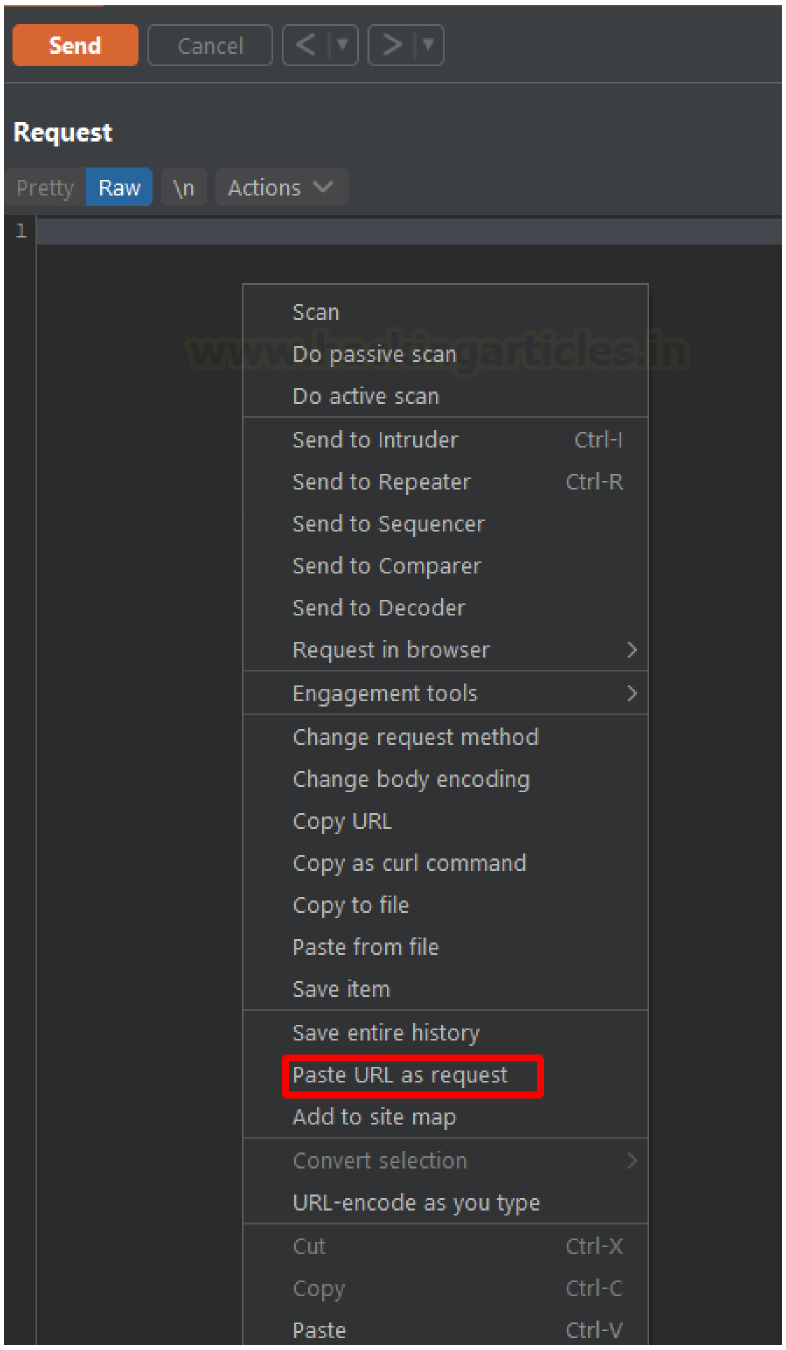
In a Penetration testing environment, there are situations where it is required to test the response of a particular URL without actually capturing the request. Or there can be a scenario where you were able to get a particular finding but you don't have the request in the response. So, you will have to go to the HTTP History and locate that particular request and send it to the repeater. This process can be shortened by just using the URL. Here, we are copying the URL from our Web Browser.



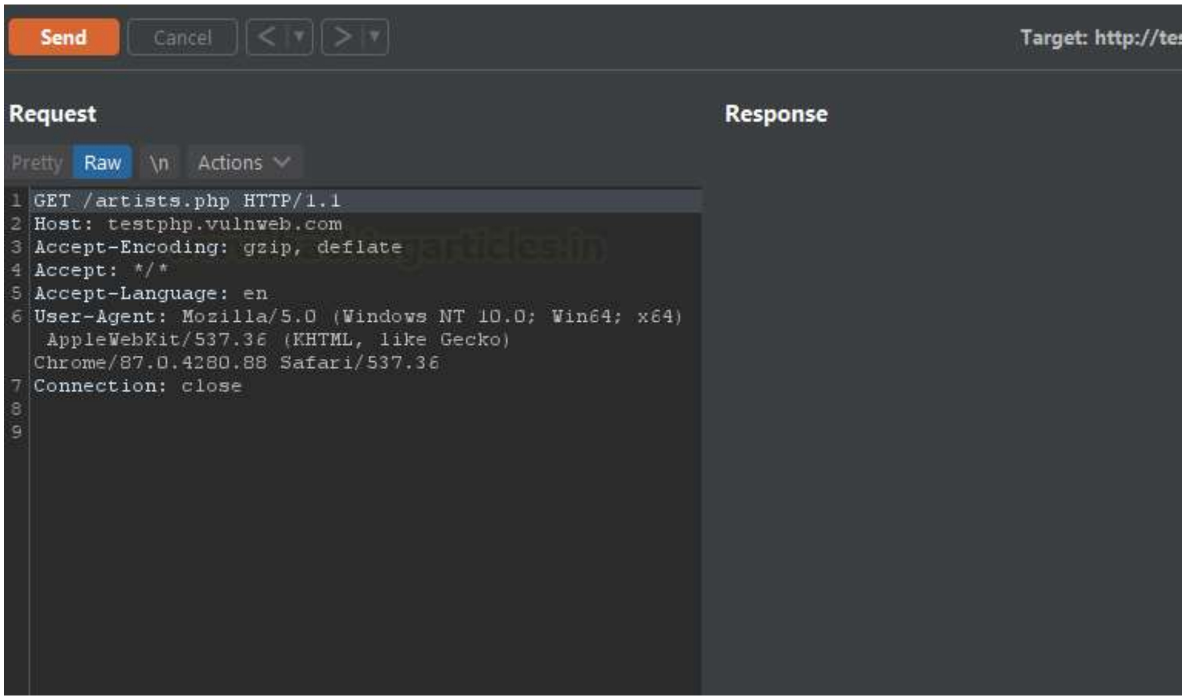
Now we are going to the repeater and we will be creating a new tab by right-clicking on the section. We will be asked if the new tab is supposed to be an HTTP request or a Web Socket request. Since we are trying to work with a Webpage, we choose HTTP.



Now, we right-click on the empty Request section and choose the “Paste URL as request” option from the dropdown menu.



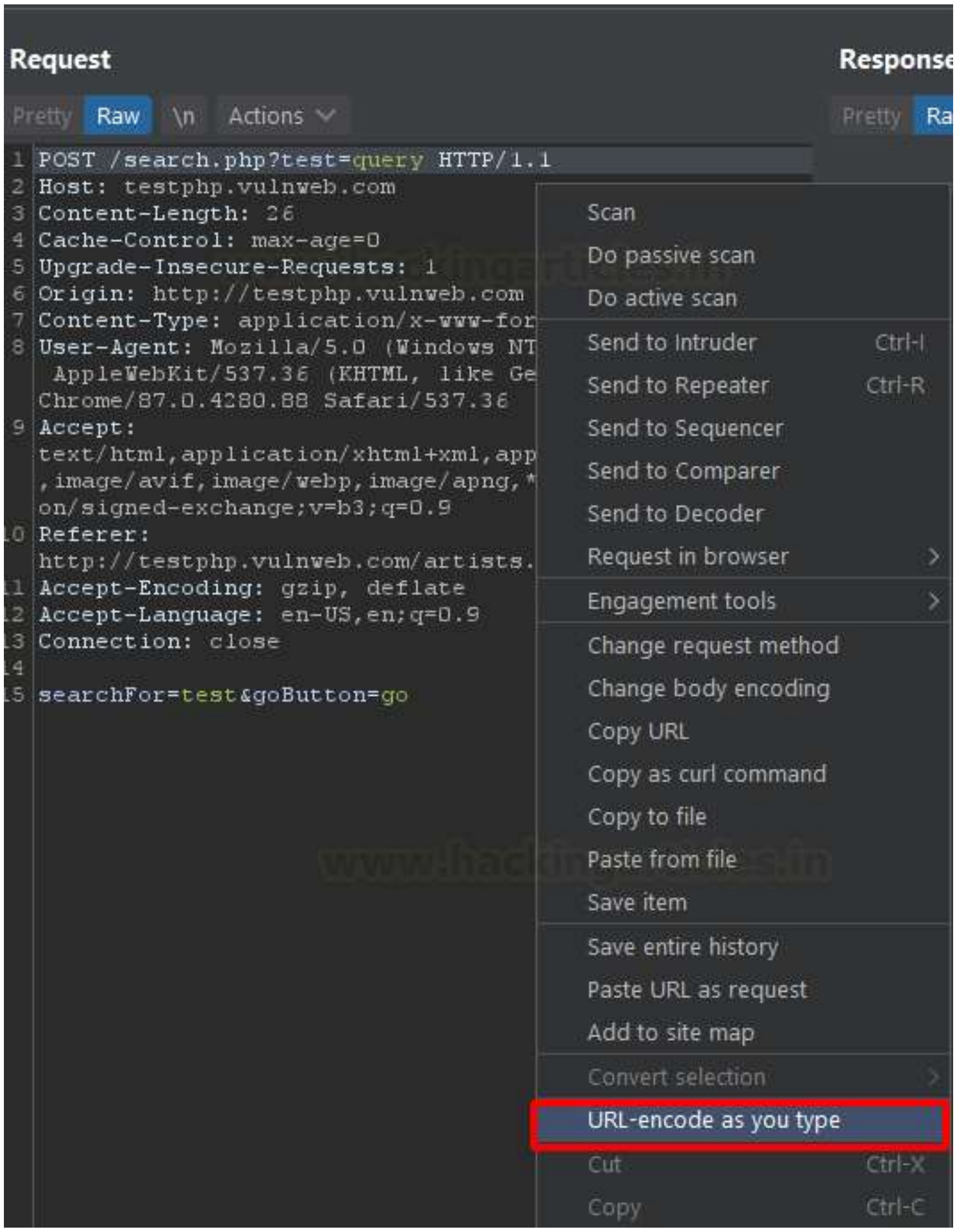
We can see from the image below that the URL was converted into a proper request with all the basic header added to it by the Burp. This is done automatically. Now that we have the request for the URL that we want to investigate, we can just click on the Send button and observe the response generated by it.



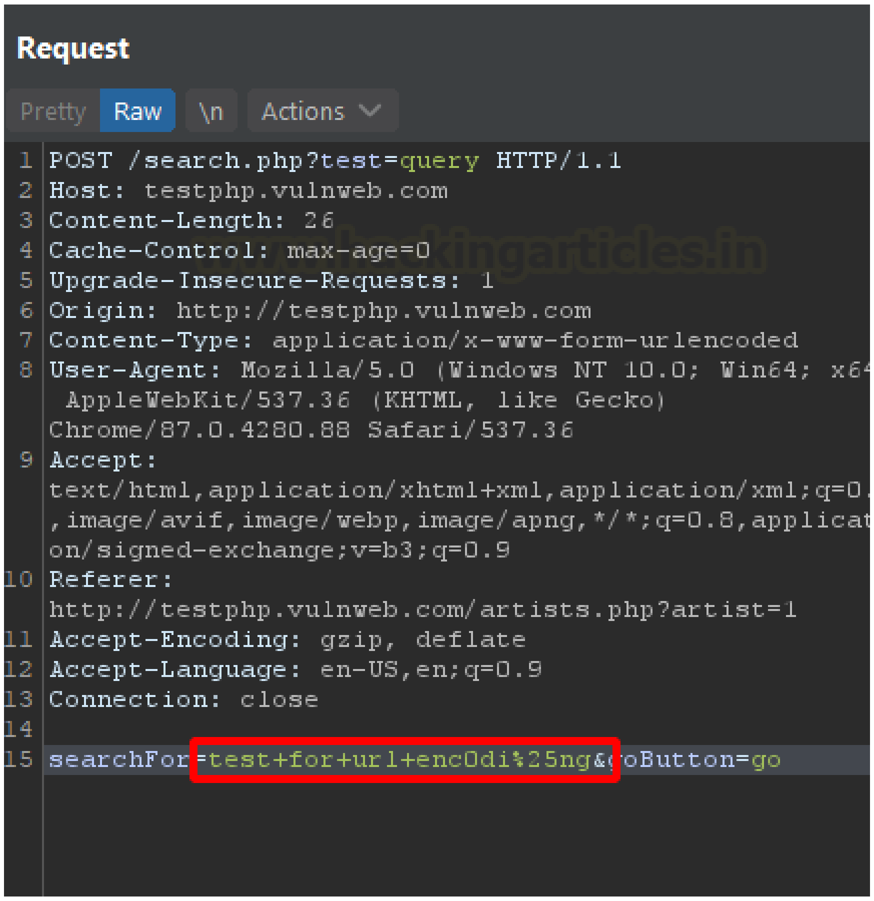
```
GET /artists.php HTTP/1.1
Host: testphp.vulnweb.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Connection: close
```

## URL Encode

Web servers don’t deal with the spaces and certain symbols easily. Hence, the space and certain symbols such as the & are encoded in URL Encode format. Developers program the websites to encode the data Client side before generating the request and then sending it to the webserver. Now, when making changes to the request inside the Repeater, you forget to provide the proper encoding the request may behave in an unpredicted way. Also, some filters are placed to find certain symbols such as < > but the URL may Encode equivalent of those are not filtered and that filter can be bypassed by just encoding them. In the demonstration below, we have a request inside the Repeater and we right-click on the request and choose the URL-encode as you type option. This will encode our text to URL Encode format as we type without any effort from our end.


```
POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 26
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-for
User-Agent: Mozilla/5.0 (Windows NT
 AppleWebKit/537.36 (KHTML, like Ge
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,app
,image/avif,image/webp,image/apng,*
on/signed-exchange;v=b3;q=0.9
Referer:
http://testphp.vulnweb.com/artists.
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

searchFor=test&goButton=go
```

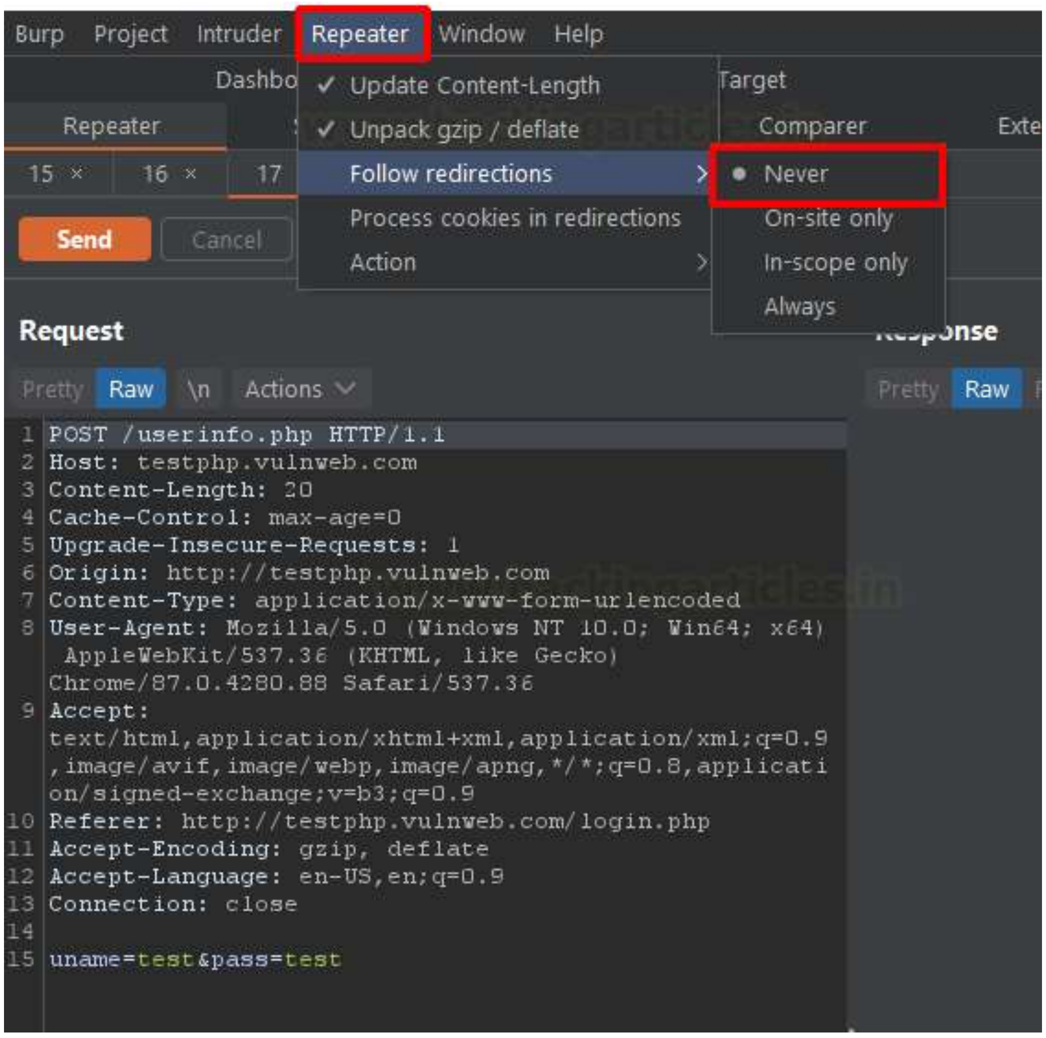
We can see that we have added the values to the searchFor parameter and the spaces are converted into + and the entire string has been converted into URL Encode format. This can be disabled by choosing the same option from the menu.



```
POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 26
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x6
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.
,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
on/signed-exchange;v=b3;q=0.9
Referer:
http://testphp.vulnweb.com/artists.php?artist=1
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

searchFor=test+for+url+enc0di%25ng&goButton=go
```

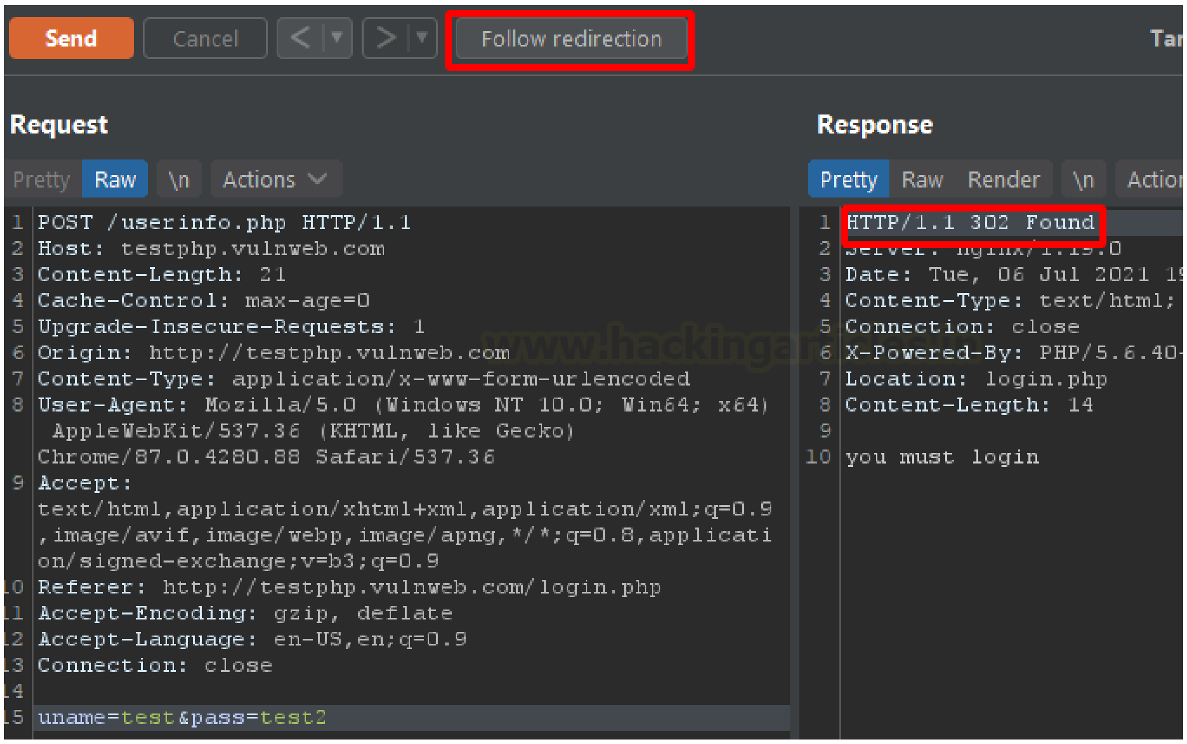
## Following Redirection

Redirection is an important part of any web application. This can help the user navigate the web pages in a way that the developer intended. It can also help the developer to integrate multiple different web applications into a single website. Web Browsers follow the redirection by default and it poses a problem to a penetration tester because while testing certain scenarios such as Open Redirection or Web Cache Poisoning, the penetration tester needs to toggle and observe the redirection response from the application. The Repeater has an option that can help with this or similar situations. It provides the user with an option to either never to follow redirections or follow the redirection for the on-site redirection or follow the redirection for In-scope domains only or to just Always Follow the Redirection. This provides the user to configure the Repeater as per their requirement that is based on the build of the application.



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 20
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
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/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

uname=test&pass=test
```

Since, in the previous step, we choose Never to follow the redirections, we see a 302 response in the image below. Now that we have a Redirection response, to move forward with the request, we can click on the Follow redirection button as shown in the image below to continue.

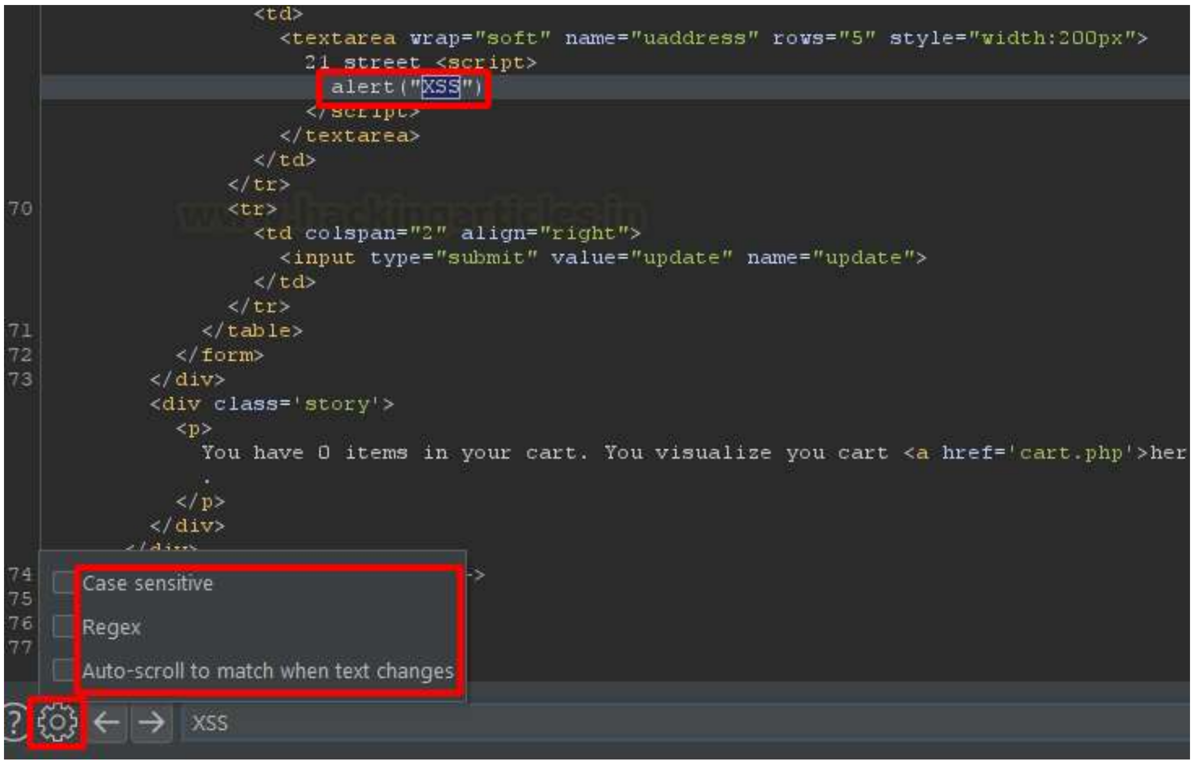


```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 21
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

uname=test&pass=test2
```

```
HTTP/1.1 302 Found
Server: nginx/1.19.0
Date: Tue, 06 Jul 2021 1
Content-Type: text/html;
Connection: close
X-Powered-By: PHP/5.6.40-
Location: login.php
Content-Length: 14

you must login
```

## Search

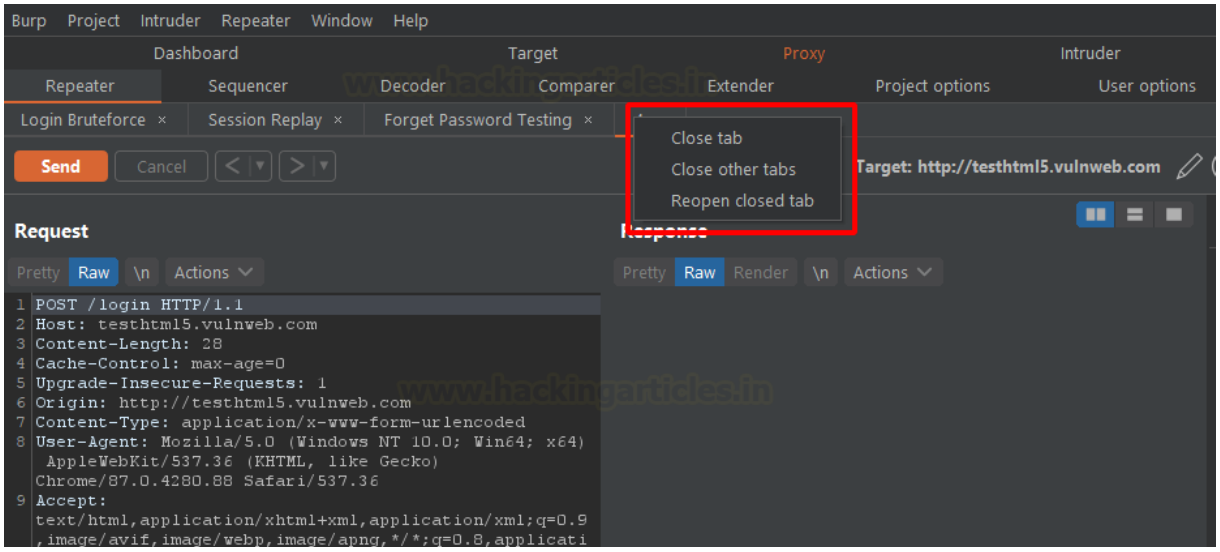
In both sections i.e., Request and Response of the Repeater, we have a search bar at the bottom. Since the response would contain all the HTML code to create the webpage, it tends to be very lengthy and it can be very difficult to look for any particular keyword in the code. This is a problem when we are trying to see if any parameters that we passed were reflected in the response or not. The Search is configured to automatically move to the first keyword it finds and the Left and Right arrow keys can be used to toggle between the different occurrences of the keyword. Some settings can be toggled while using the Search. We can check the box to make the search Case Sensitive or we can use a Regex value to search for a general type of occurrence. At last, the Auto-scroll can be triggered each time any change be made to the Search bar. This can also help look for particular keywords quickly.



```
<td>
  <textarea wrap="soft" name="uaddress" rows="5" style="width:200px">
    21 street <script>
      alert("XSS")
    </script>
  </textarea>
</td>
</tr>
<tr>
  <td colspan="2" align="right">
    <input type="submit" value="update" name="update">
  </td>
</tr>
</table>
</form>
</div>
<div class='story'>
  <p>
    You have 0 items in your cart. You visualize you cart <a href='cart.php'>her
    .
  </p>
</div>
```

## Reopening Closed Tab

If you have been using Burp for some time now, you might have accidentally closed the Repeater Tab that you don't want to close. You are not alone and it is not your fault. The Close (X) is placed in such a way that it is very possible to close the tab without any intention to do so. After numerous requests to the PortSwigger, They made this option where you can Right Click on the location where all the tabs lie and Choose The Reopen Closed Tab option to open any tab that you closed accidentally.

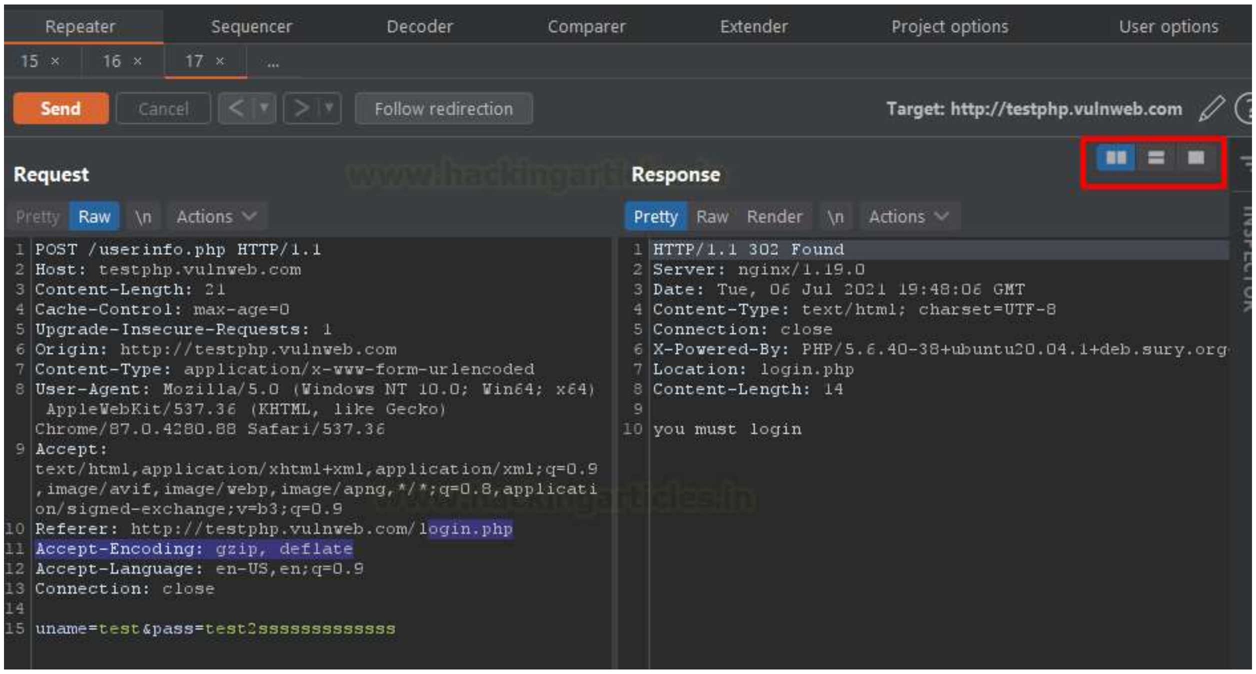


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder
Repeater Sequencer Decoder Comparer Extender Project options User options
Login Bruteforce × Session Replay × Forget Password Testing ×
Close tab
Close other tabs
Reopen closed tab
Send Cancel
Target: http://testhtml5.vulnweb.com
Request
Response
Pretty Raw \n Actions
Pretty Raw Render \n Actions

```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Content-Length: 28
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testhtml5.vulnweb.com
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/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
```

## Views

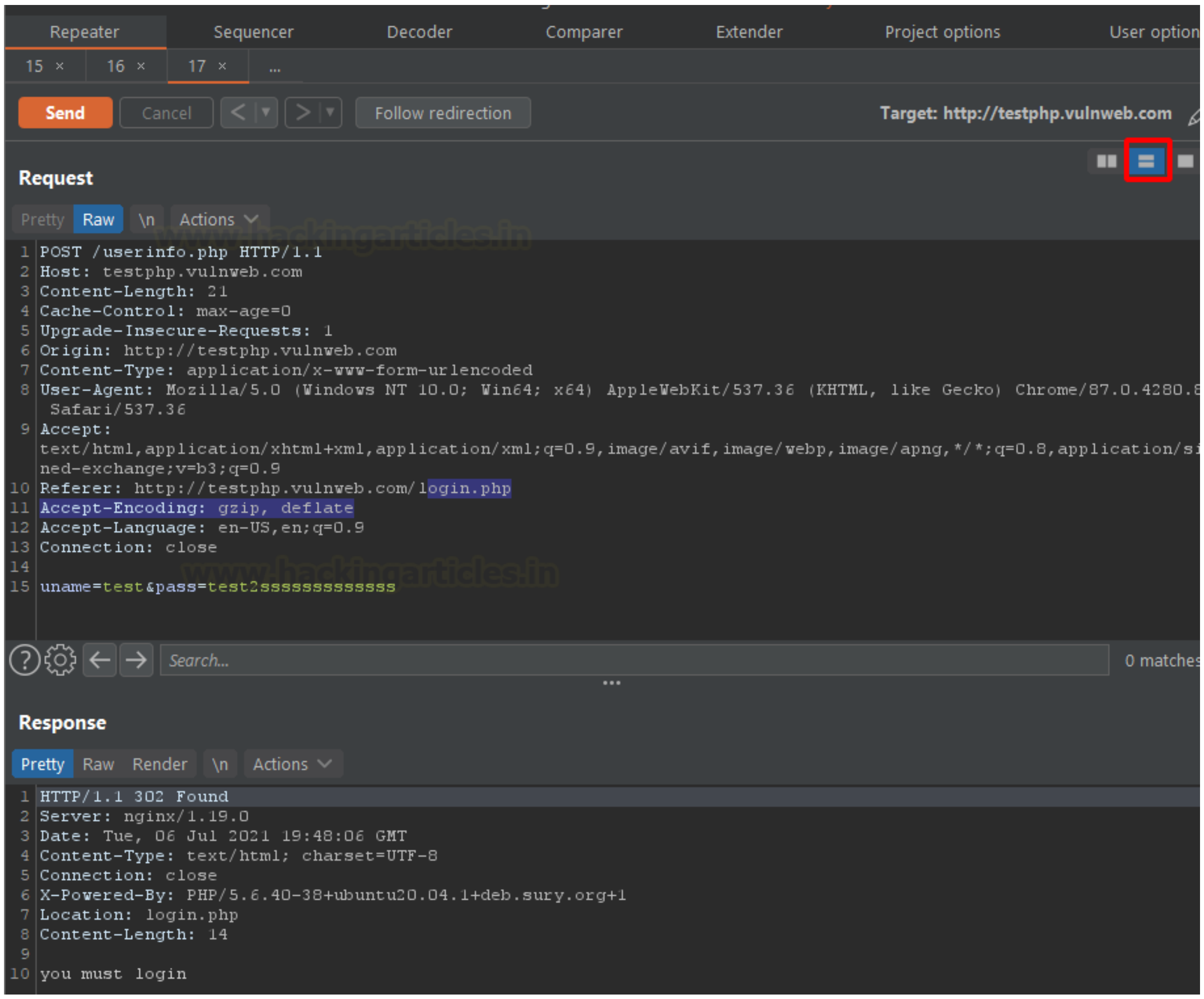
Multiple views are provided to view the Request and the Response sections in the Burp Suite. This pretty much sums based on the user's preference. The three options that are provided are the classic Side by Side Panels as shown in the image below.



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 21
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
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/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

uname=test&pass=test2sssssssssssss
```

```
HTTP/1.1 302 Found
Server: nginx/1.19.0
Date: Tue, 06 Jul 2021 19:48:06 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org
Location: login.php
Content-Length: 14

you must login
```

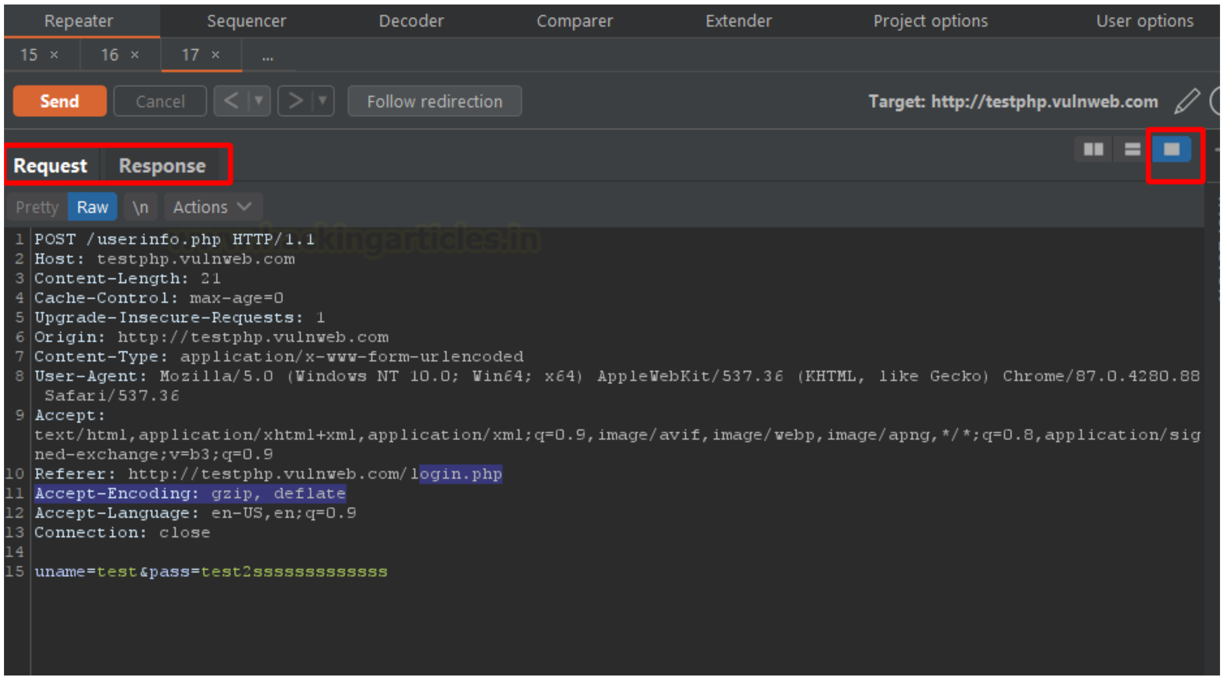
Next, we have the Top-Bottom Approach. This can be useful if any user is comfortable with this orientation.



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 21
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
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/87.0.4280.8
 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/si
ned-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

uname=test&pass=test2ssssssssssss
```

```
HTTP/1.1 302 Found
Server: nginx/1.19.0
Date: Tue, 06 Jul 2021 19:48:06 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Location: login.php
Content-Length: 14

you must login
```

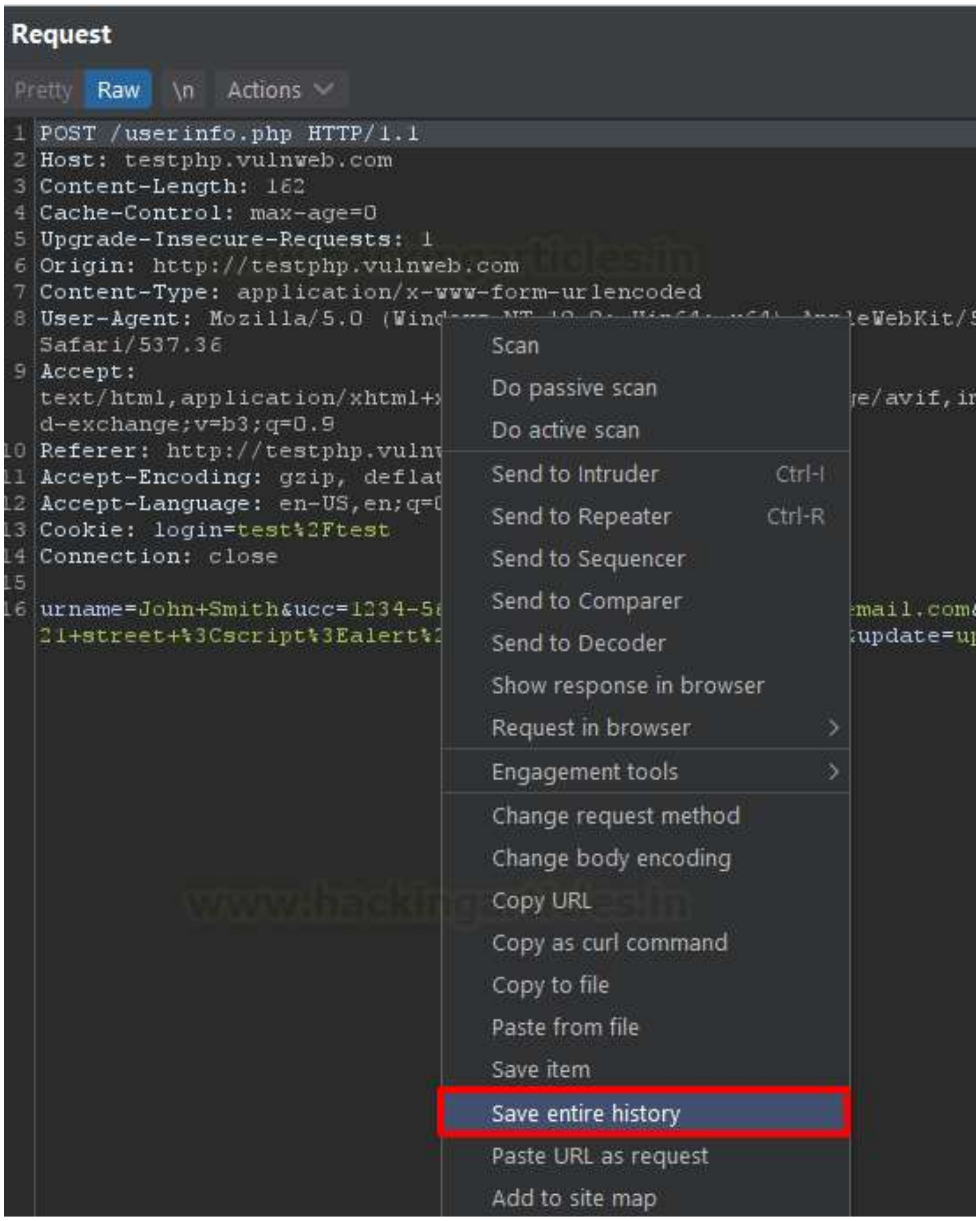
Finally, we have the Tabs approach that adds another set of tabs to the Burp's Tab scheme. It gives the user a complete area with a particular section. This can come in handy with certain scenarios. It provides the Tabs for Request and Response and we can click on those to access both of them.



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 21
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
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/87.0.4280.88
 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

uname=test&pass=test2sssssssssssss
```

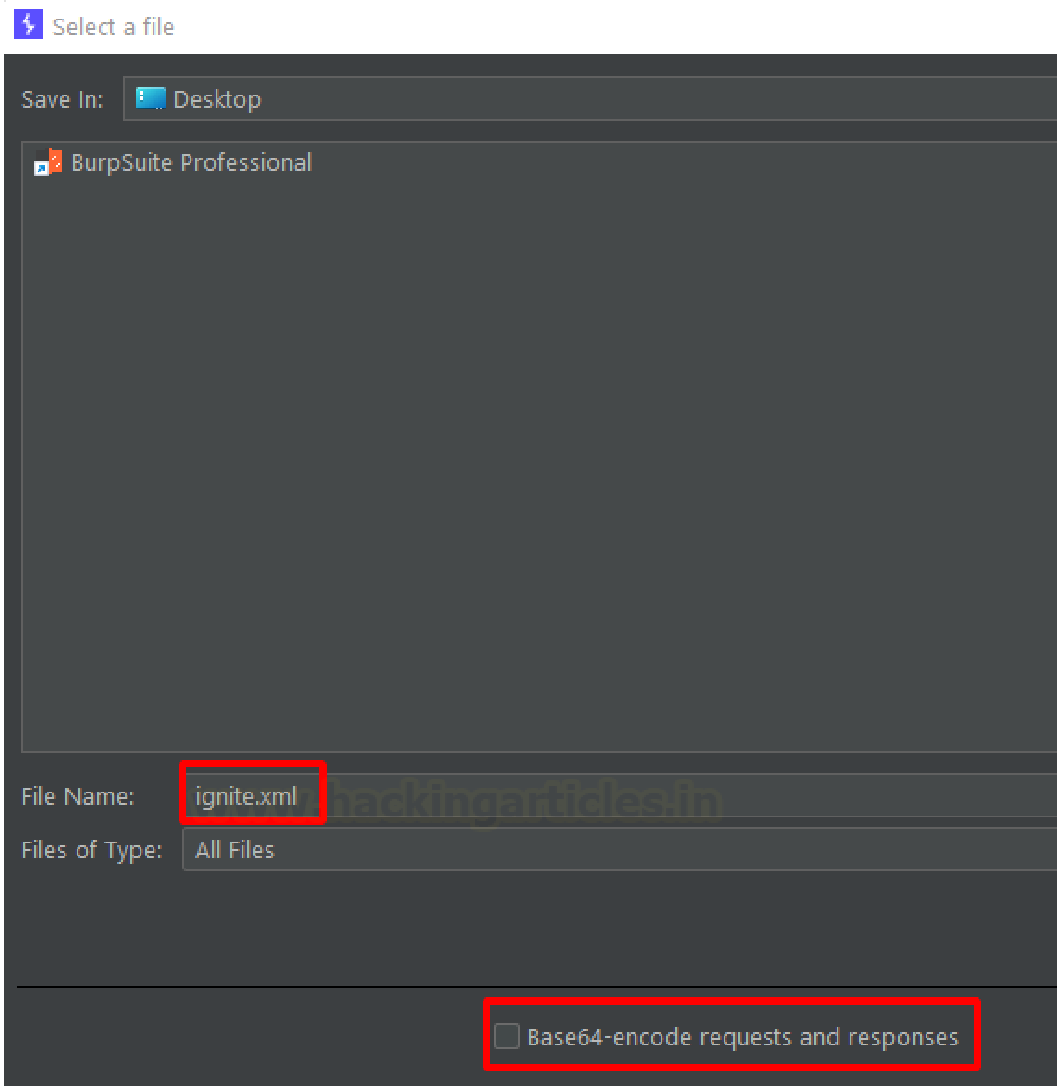
## Exporting Repeater Data

As we know that documentation is a vital part of any engagement. As we send multiple requests to form the repeater and with each request, we tend to make some changes to the request. So, to keep a track of it outside of Burp, we have the feature that allows us to save the history of all the requests sent. After working with multiple requests, we right-click on the request and choose the “Save entire history” option from the drop-down menu.



```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 162
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Wind                        eWebKit/S
Safari/537.36
Accept:
text/html,application/xhtml+x                        e/avif,ir
d-exchange;v=b3;q=0.9
Referer: http://testphp.vulnw
Accept-Encoding: gzip, deflat
Accept-Language: en-US,en;q=0
Cookie: login=test%2Ftest
Connection: close

urname=John+Smith&ucc=1234-56                        mail.com&
21+street+%3Cscript%3Ealert%2                        update=up
```

This opens up a window that asks us to provide the location of the file that we want to save. Upon browsing the location, we name the file as per our requirement and provide the extension as XML. We can also encode the requests in Base64 as well. However, in this case, we are not doing so.


Select a file

Save In: Desktop

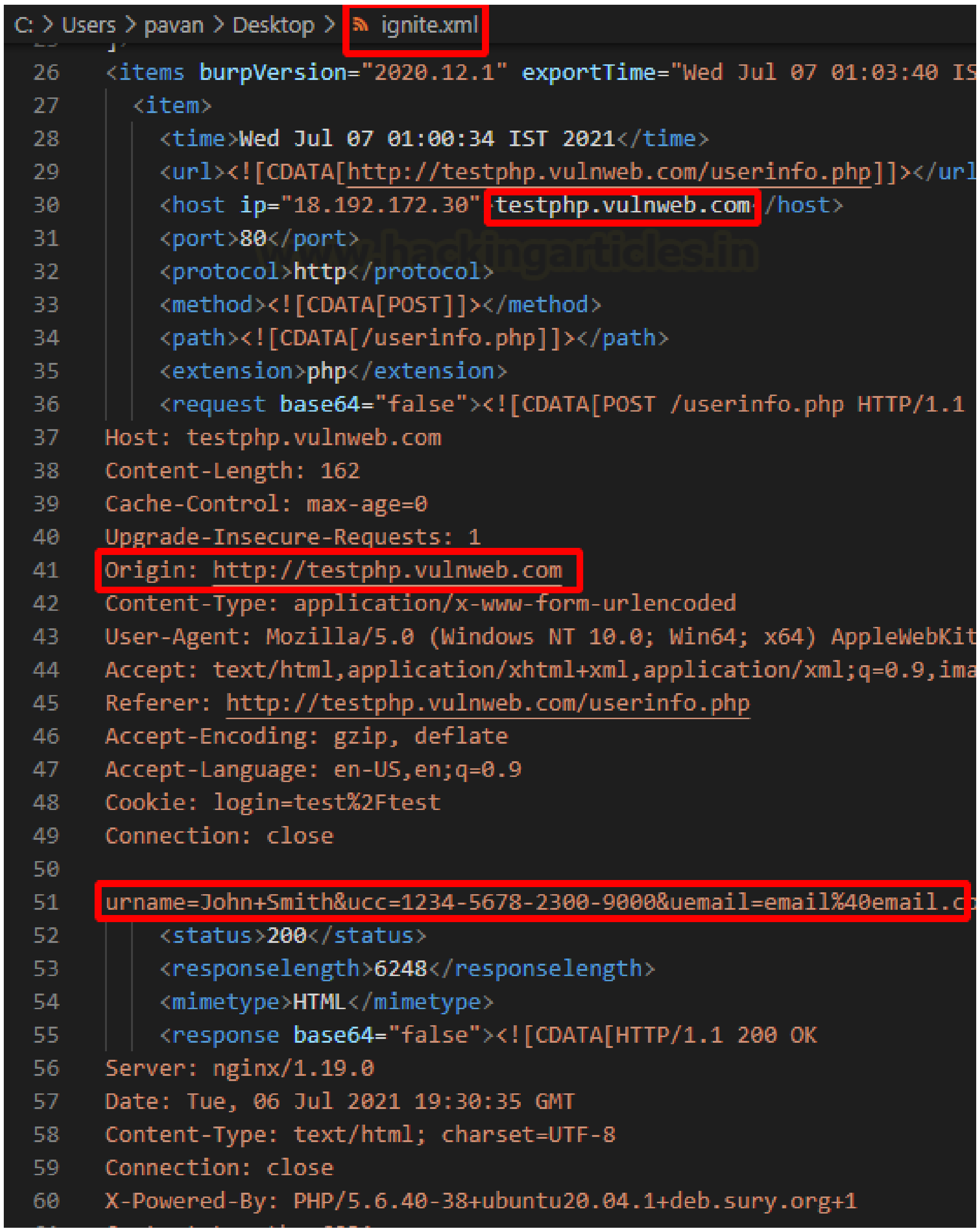
BurpSuite Professional

File Name: ignite.xml

Files of Type: All Files

[ ] Base64-encode requests and responses

Now, the history of all the requests that were sent will be found at the provided location in the XML format. In the image provided below we can see that we have the IP Address of the target, Domain of target, Port of the target, Protocol used, and the complete contents of the Request and the Response.



```
C: > Users > pavan > Desktop > ignite.xml
<items burpVersion="2020.12.1" exportTime="Wed Jul 07 01:03:40 IS
  <item>
    <time>Wed Jul 07 01:00:34 IST 2021</time>
    <url><![CDATA[http://testphp.vulnweb.com/userinfo.php]]></url
    <host ip="18.192.172.30">testphp.vulnweb.com</host>
    <port>80</port>
    <protocol>http</protocol>
    <method><![CDATA[POST]]></method>
    <path><![CDATA[/userinfo.php]]></path>
    <extension>php</extension>
    <request base64="false"><![CDATA[POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 162
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,ima
Referer: http://testphp.vulnweb.com/userinfo.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: login=test%2Ftest
Connection: close

uname=John+Smith&ucc=1234-5678-2300-9000&uemail=email%40email.c
    <status>200</status>
    <responselength>6248</responselength>
    <mimetype>HTML</mimetype>
    <response base64="false"><![CDATA[HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Tue, 06 Jul 2021 19:30:35 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
```

## Conclusion

The repeater is one of the basic tools of the BurpSuite. However, through extensive research and developments, a lot of hidden features are added to the tool. It has reached a point where many of these features can ease the life of any Pentester.