

BurpSuite

Burp Suite is an integrated platform/graphical tool for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp Suite is a fully featured web application attack tool: it does almost anything that you could ever want to do when penetration testing a web application.

It is the most popular tool among professional web app security researchers and bug bounty hunters. Its ease of use makes it a more suitable choice over free alternatives like OWASP ZAP.

The tools offered by BurpSuite are:

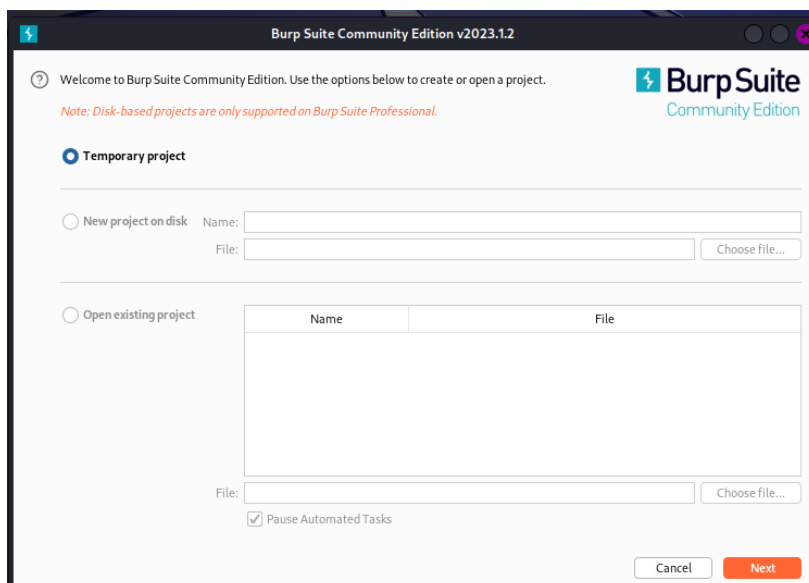
- Spider
- Proxy
- Intruder
- Repeater
- Sequencer
- Decoder
- Extender
- Scanner

bWAPP, or a buggy web application, is a free and open source deliberately insecure web application. It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.

bWAPP prepares one to conduct successful penetration testing and ethical hacking projects. It has over 100 web vulnerabilities! It covers all major known web bugs, including all risks from the OWASP Top 10 project.

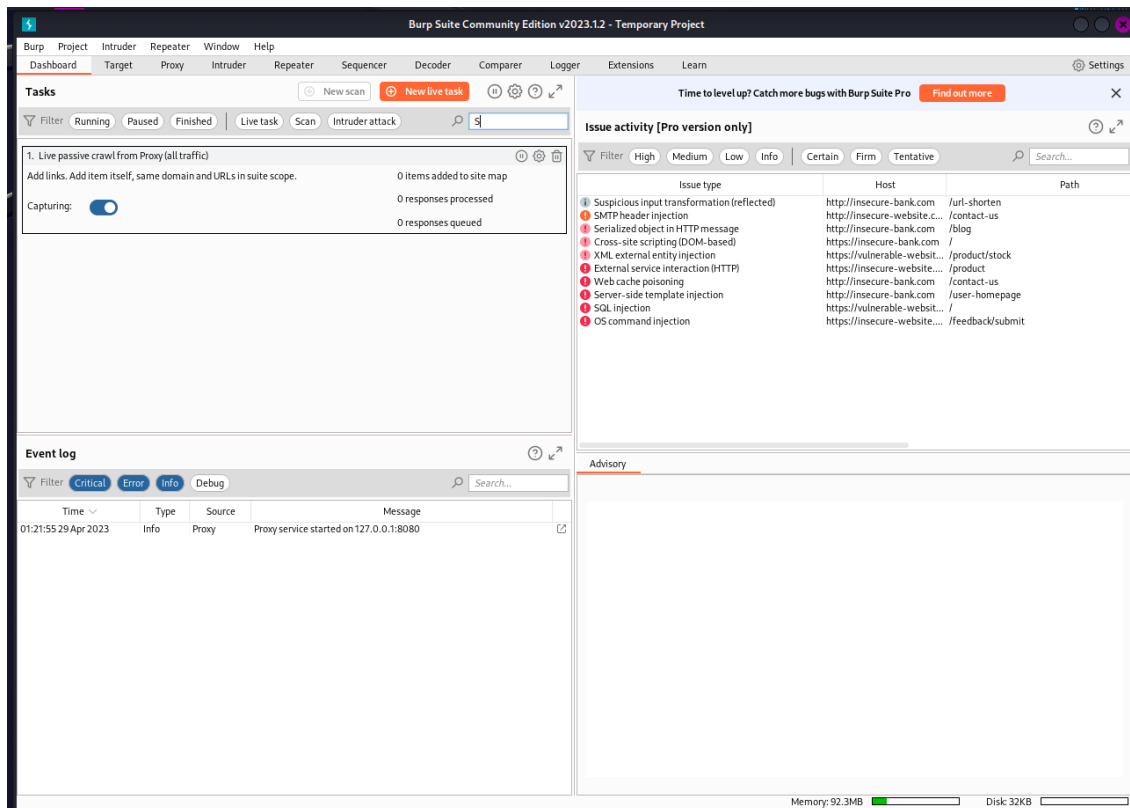
So, we will use bWAPP as our target for burp suite.

So, the homepage of burpsuite:

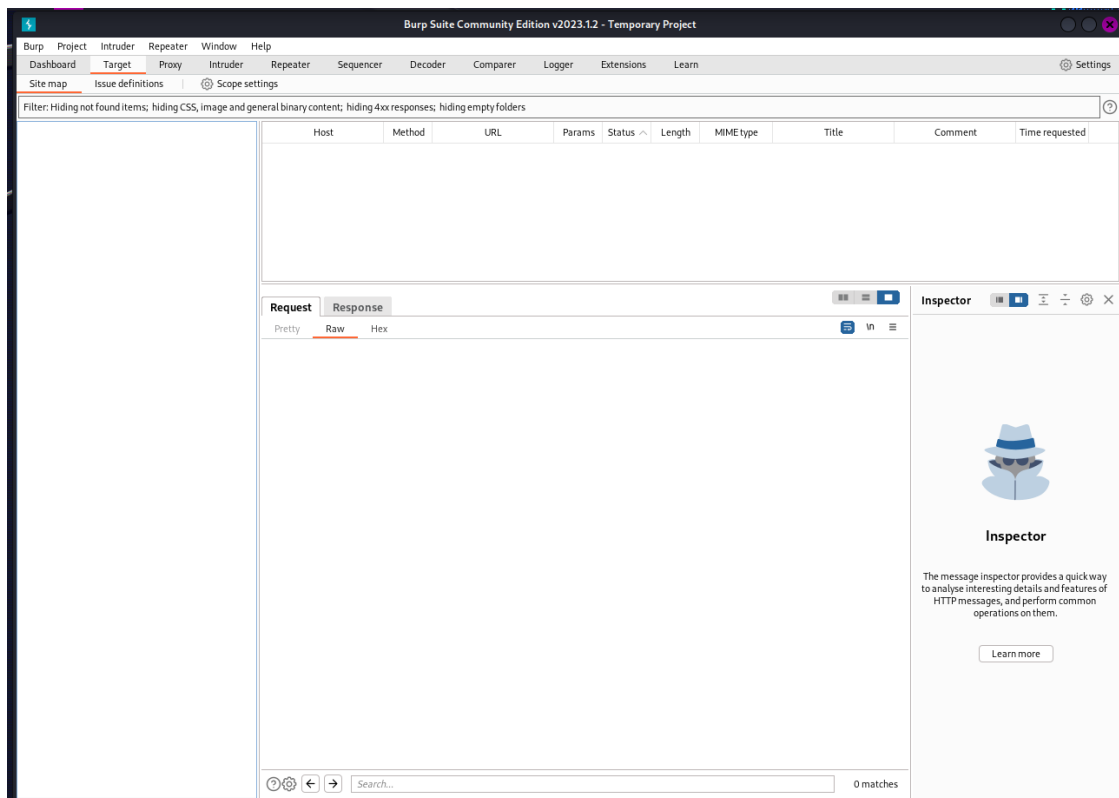


So, in community version we can only create temporary projects. If we want it permanent, we have to buy the professional version.

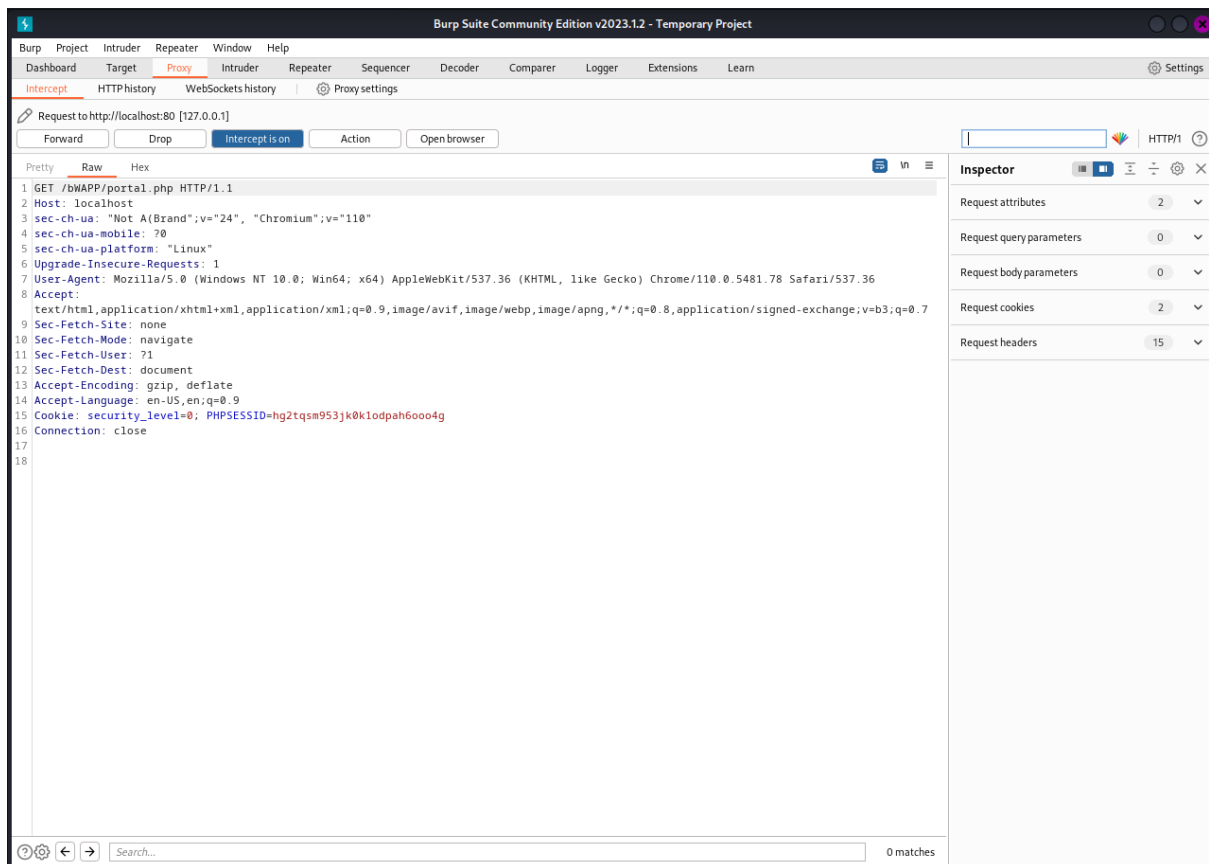
After we create the project, we will land on this screen.



Here is the dashboard, it will give some basic information.

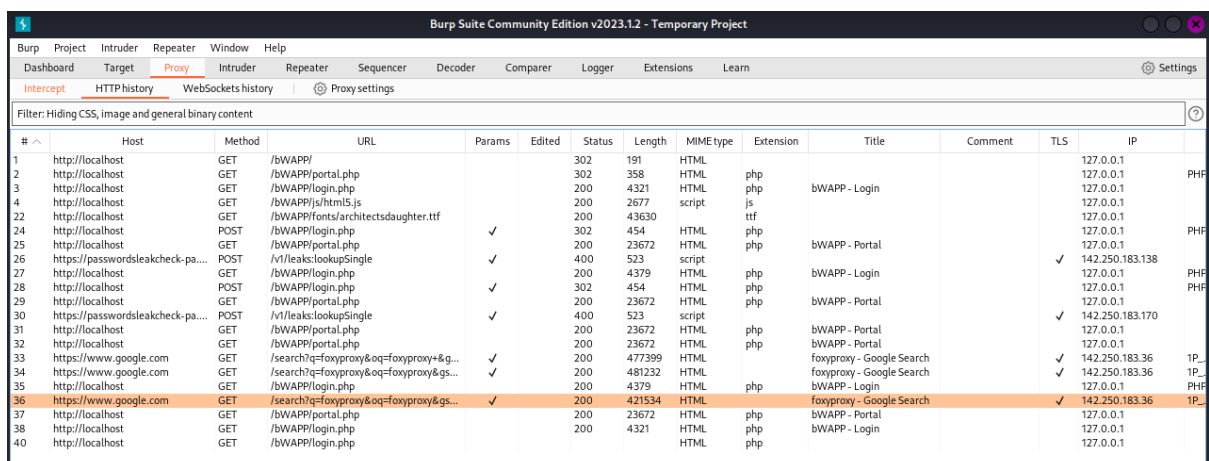


In target we give the information we want to work with, it will check any website the browser is hooked, it will look for every website your browser is running.

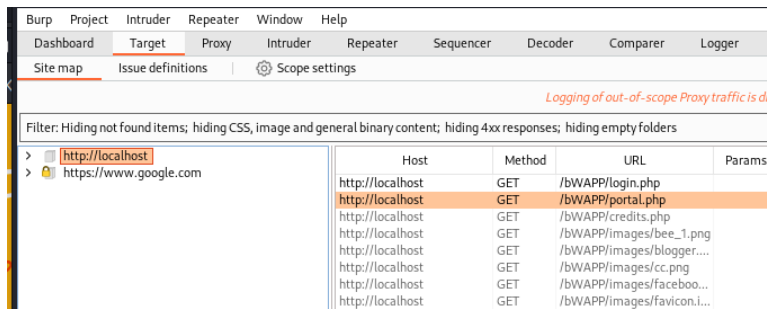


In proxy tab, the interceptor will grab every request the browser makes, because my browser is already hooked to burp suite.

Now I have made a lot of requests, so all the request will be present in history tab.

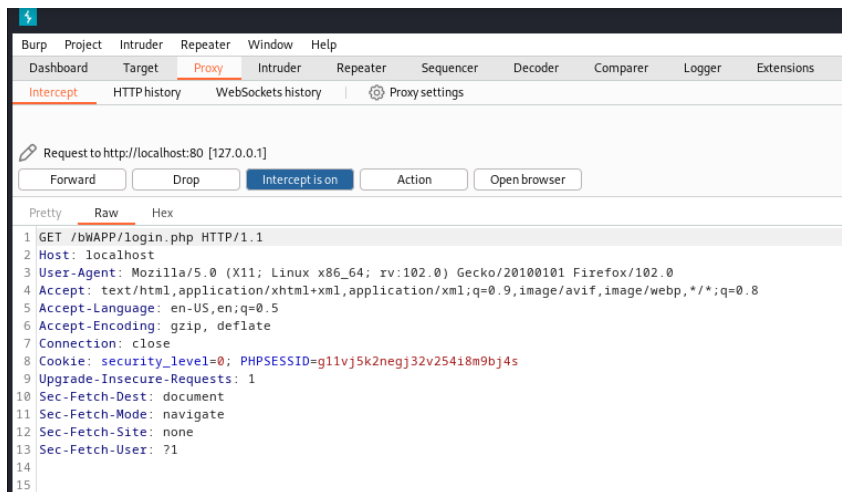


Now I added the localhost request in 'add to scope' so it will filter the remaining traffic in my history.



Now I have request anything from the browser it will go through burpsuite.

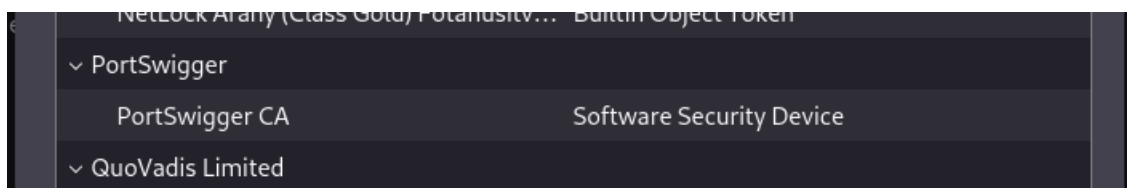
And we have the option whether to forward or drop.



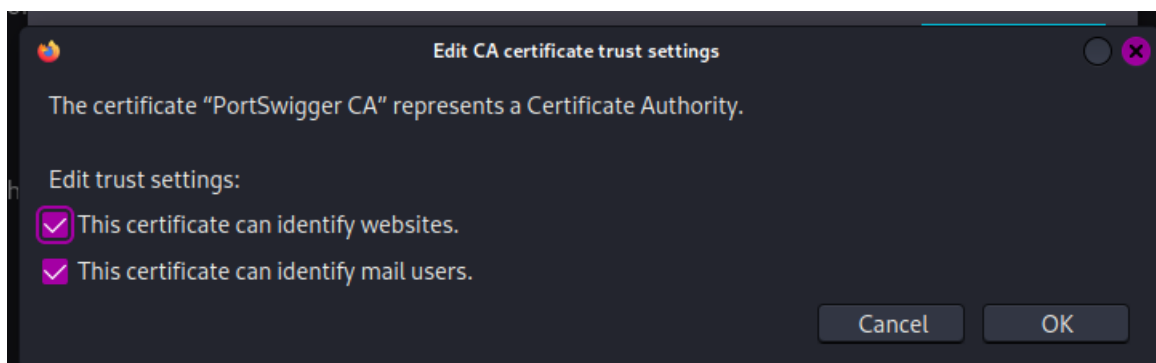
So, if I forward the website will long and if I drop the website won't load.

We can intercept the https request too, but for that we need to install a certificate available on 'http://burpsuite/'.

After downloading we just need to add that certificate into our browser, and then we can access https request too.



And make sure this is checked:



Here the history is too much messy, so ill clear history by using 'clear history' button when we right click.

30	https://passwordsleakcheck-pa...	POST	/v1/leaks:lookupSingle	✓
31	http://localhost	GET	/bWAPP/portal.php	
32	http://localhost	GET	/bWAPP/portal.php	
33	https://www.google.com	GET	/search?	
34	https://www.google.com	GET	/search?	http://localhost/bWAPP/portal.php
35	http://localhost	GET	/bWAPP	Remove from scope
36	https://www.google.com	GET	/search?	Scan
37	http://localhost	GET	/bWAPP	Send to Intruder Ctrl+I
38	http://localhost	GET	/bWAPP	Send to Repeater Ctrl+R
40	http://localhost	GET	/bWAPP	Send to Sequencer
42	http://localhost	GET	/bWAPP	Send to Comparer (request)
43	http://localhost	GET	/bWAPP	Send to Comparer (response)
44	http://localhost	GET	/bWAPP	Show response in browser

Request

Pretty Raw Hex

```

1 GET /search?q=foxyproxy&oq=foxyproxy&gs_l=
UTF-8 HTTP/2
2 Host: www.google.com
3 Sec-Ch-Ua: "Not A(Brand";v="24", "Chromium
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0;
Chrome/110.0.5481.78 Safari/537.36
8 Accept:
text/html,application/xhtml+xml,application
signed-exchange;v=b3;q=0.7
9 X-Client-Data: CNHyygE=
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document

```

Request in browser >

Engagement tools [Pro version only] >

Show new history window

Add comment

Highlight >

Delete item

Clear history

Copy URL

Copy as curl command

Copy links

Save item

Proxy history documentation

On the left-hand side are the request my browser was making and on the right-hand side were the response from the server.

The screenshot shows the Burp Suite interface. The top panel displays a list of intercepted requests. The bottom panel shows a detailed view of a request and response.

Request:

```

1 GET /bWAPP/portal.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20180813 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Encoding: gzip, deflate
6 Referer: http://localhost/bWAPP/login.php
7 Connection: close
8 Cookie: security_level=0; PHPSESSID=bv234p43jU91456vumeig98r
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1

```

Response:

```

1 HTTP/1.1 200 OK
2 Date: Sun, 30 Apr 2023 05:28:28 GMT
3 Server: Apache/2.4.18 (Ubuntu)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate
6 Pragma: no-cache
7 Vary: Accept-Encoding
8 Content-Length: 23369
9 Connection: close
10 Content-Type: text/html; charset=UTF-8
11
12 <!DOCTYPE html>
13 <html>
14
15 <head>
16
17 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
18
19 <!--<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Architects-Daughter">-->
20 <link rel="stylesheet" type="text/css" href="stylesheets/stylesheet.css" media="screen" />
21 <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />
22
23 <!--script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script-->
24 <script src="js/html5.js">
25 </script>
26
27 <title>

```

If we click on any websites the request is send to the web server of that website. And the web server will response you with the website you requested.

Request:

Request	
	Pretty Raw Hex
1	GET /bWAPP/portal.php HTTP/1.1
2	Host: localhost
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5	Accept-Language: en-US,en;q=0.5
6	Accept-Encoding: gzip, deflate
7	Referer: http://localhost/bWAPP/login.php
8	Connection: close
9	Cookie: security_level=0; PHPSESSID=bv234p43ju391456vumaigk98r
10	Upgrade-Insecure-Requests: 1
11	Sec-Fetch-Dest: document
12	Sec-Fetch-Mode: navigate
13	Sec-Fetch-Site: same-origin
14	Sec-Fetch-User: ?1
15	

- So, we can see I requested the bWAPP page in GET method.
- The host I'm using is localhost.
- I'm using mozilla firefox to request.
- What type of acceptable input it allows.
- The referrer is from where the request comes from.
- The connection is close because it's just a simple static page.
- And much more...

Response	
	Pretty Raw Hex Render
1	HTTP/1.1 200 OK
2	Date: Sun, 30 Apr 2023 05:28:28 GMT
3	Server: Apache/2.4.55 (Debian)
4	Expires: Thu, 19 Nov 1981 08:52:00 GMT
5	Cache-Control: no-store, no-cache, must-revalidate
6	Pragma: no-cache
7	Vary: Accept-Encoding
8	Content-Length: 23369
9	Connection: close
10	Content-Type: text/html; charset=UTF-8
11	
12	<!DOCTYPE html>
13	<html>
14	
15	<head>
16	
17	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
18	
19	<!--<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Architects+Daughter">-->
20	<link rel="stylesheet" type="text/css" href="stylesheets/styleSheet.css" media="screen" />
21	<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />
22	
23	<!--<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>-->
24	<script src="js/html5.js">
25	</script>
26	
27	<title>
28	bWAPP - Portal
29	</title>
30	
31	</head>
32	<body>
33	
34	<header>
35	
36	<h1>
37	bWAPP
38	</h1>
39	
40	<h2>
41	an extremely buggy web app !
42	</h2>

- So, in response we got 200 which we the server didn't get any error in the request and the server will forward us the page.
- It will return the date when it happened.
- What server was used to response to my request.
- When will it expires.
- Content type and its length.
- And the whole web page source code. We can see the source code from inspect tool also but we won't see the header information.
- And much more...

Now I have cleared the history of burpsuite. And I once again went to login page of bWAPP and login. So in the history tab we will see a post method.

Note the security level is low i.e., zero. We can change it to medium (1) and high (2).

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extensions									
Intercept HTTP history WebSockets history Proxy settings									
Filter: Hiding CSS, image and general binary content									
#	Host	Method	URL	Params	Edited	Status	Length	M	
27	http://localhost	GET	/bWAPP/login.php			200	4321	HT	
28	http://localhost	POST	/bWAPP/login.php	✓		302	454	HT	
29	http://localhost	GET	/bWAPP/portal.php			200	23672	HT	
30	http://localhost	GET	/bWAPP/login.php			200	4379	HT	
32	http://localhost	GET	/bWAPP/js/html5.js			200	2677	scr	
47	http://localhost	GET	/bWAPP/fonts/architectsdaughter.ttf			200	43630		

Post request is like im sending data to be computed by the application. i.e., I'm sending username and password on the login page.

And we can see the username and password in burpsuite.

```

11 Referer: http://localhost/bWAPP/login.php
12 Cookie: security_level=0; PHPSESSID=bv234p43ju391456vumaigk98r
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 login=bee&password=bug&security_level=0&form=submit

```

In the request section of burpsuite we can see the login id and password i.e., bee and bug, also the security level which was low meaning 0. And we submit the form.

Now if I want to change the once again sends the same request. I can go to the website logout myself change the setting and login again.

Or I can send the request we already send to repeater which is a tool in burpsuite.

#	Host	Method	URL	Params	Ed
58	http://localhost	GET	/bWAPP/login.php		
59	http://localhost	POST	/bWAPP/login.php		
60	http://localhost	GET	/bWAPP/porta	http://localhost/bWAPP/login.php	
Remove from scope					
Scan					
Send to Intruder Ctrl+I					
Send to Repeater Ctrl+R					

And now we will go to the repeater tab and see the same request.

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Exten
1 x +
Send Cancel < >
Request
Pretty Raw Hex
1 POST /bWAPP/login.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 51
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/bWAPP/login.php
12 Cookie: security_level=2; PHPSESSID=2aalvr6p5j1ss23ihv7rg8c33i
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 login=bee&password=bug&security_level=0&form=submit

If we send the request, it will response.

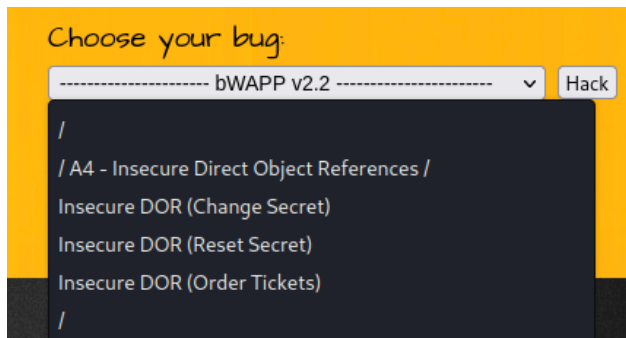
Response
Pretty Raw Hex Render
1 HTTP/1.1 302 Found
2 Date: Sun, 30 Apr 2023 06:31:06 GMT
3 Server: Apache/2.4.55 (Debian)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate
6 Pragma: no-cache
7 Set-Cookie: PHPSESSID=b8qa0s6geunt4ibavb410skr14; path=/
8 Set-Cookie: security_level=0; expires=Mon, 29 Apr 2024 06:31:06 GMT; Max-Age=31536000; path=/
9 Location: portal.php
10 Content-Length: 0
11 Connection: close
12 Content-Type: text/html; charset=UTF-8
13
14

It has response with 302, because we are already logged in.

We can change and manipulate the information and send again if we want in repeater mode and check what's the change in the response.

And in repeater we can have many tabs. i.e., we can do many repeater together using different tabs.

Now we are going to use one of the bug in the bWAPP website:



The Insecure DOR (Order Tickets):

We will land on this page.



Now I'll just order 1 ticket and check the burpsuite.



In burpsuite:

63	http://localhost	POST	/bWAPP/portal.php	✓	302	323	HTML	php	
64	http://localhost	GET	/bWAPP/insecure_direct_object_ref_2....		200	13664	HTML	php	bWAPP - Insecure DOR
65	http://localhost	POST	/bWAPP/insecure_direct_object_ref_2....	✓	200	13813	HTML	php	bWAPP - Insecure DOR

I visited the order ticket page. And the remaining 2 are for ordering the ticket. Simple.

Post method request:

```

9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/bWAPP/insecure_direct_object_ref_2.php
12 Cookie: security_level=0; PHPSESSID=nh16412lim6uour7i8qfih78f1
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 ticket_quantity=1&ticket_price=15&action=order

```

We have ticket quantity, price and order.

Post method response:

```

80      <p>
        You ordered <b>
          1
        </b>
        movie tickets.
      </p>
      <p>
        Total amount charged from your account automatically: <b>
          15 EUR
        </b>
        .
      </p>
      <p>
        Thank you for your order!
      </p>
81    </div>

```

It shows I purchase a ticket for 15 EUR.

Now I'll send this to repeater:

Content-Type: application/x-www-form-urlencoded	Send to Repeater	Ctrl+R
Content-Length: 46	Send to Sequencer	
Origin: http://localhost	Send to Comparer	
Connection: close	Send to Decoder	
Referer: http://localhost/bWAPP/insecure_direct_object_ref_2.php	Show response in browser	
Cookie: security_level=0; PHPSESSID=nh164121im6uour7i8qfih78f1	Request in browser	>
Upgrade-Insecure-Requests: 1	Engagement tools [Pro version only]	>
Sec-Fetch-Dest: document	Copy URL	
Sec-Fetch-Mode: navigate	Copy as curl command	
Sec-Fetch-Site: same-origin	Copy to file	
Sec-Fetch-User: ?1		

ticket_quantity=1&ticket_price=15&action=order

We have the same request. And we will send it. And I got the same response.

<pre> 12 Cookie: security_level=0; PHPSESSID=nh164121im6uour7i8qfih78f1 13 Upgrade-Insecure-Requests: 1 14 Sec-Fetch-Dest: document 15 Sec-Fetch-Mode: navigate 16 Sec-Fetch-Site: same-origin 17 Sec-Fetch-User: ?1 18 19 ticket_quantity=1&ticket_price=15&action=order </pre>	<pre> 80 <p> You ordered 1 movie tickets. </p> <p> Total amount charged from your account automatically: 15 EUR . </p> <p> Thank you for your order! </p> </pre>
--	---

Everything is working fine. Except for an Insecure direct object reference (DOR)

I should not be able to change the ticket prices. But I can. So I ordered 10 tickets for 1 EUR each.

```

11 Referer: http://localhost/bWAPP/insecure_direct_object_ref_2.php
12 Cookie: security_level=0; PHPSESSID=nh164121im6uour7i8qfih78f1
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 ticket_quantity=10&ticket_price=1&action=order

```

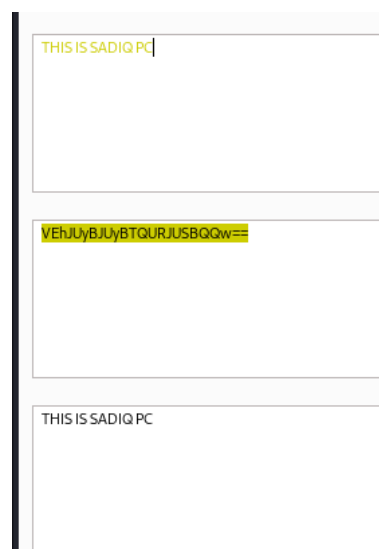
```
<p>
  You ordered <b>
    10
  </b>
  movie tickets.
</p>
<p>
  Total amount charged from your account automatically: <b>
    10 EUR
  </b>
  .
</p>
<p>
  Thank you for your order!
...
```

And in response we got 10 tickets for 10 EUR.

Here burpsuite didn't help me hack. It just allows me to see somethings I shouldn't see. And I did the manipulation in the ticket prices.

Here the developer made a mistake of letting us manipulate the prices. With the help of burpsuite and the repeater tool, we were able to access that information and we ourself change the price.

The decoder is a tool that encodes and decodes.



THIS IS SADIQ PC

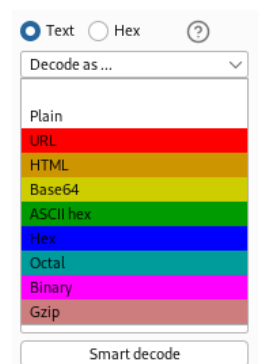
VEhJyBjYyBTQURJUSBQGW==

THIS IS SADIQ PC

Here I type something and then encode as base64. Then I got that gibberish.

Then I decode the gibberish as base64 and I got my string back.

We can encode and decode in many types:



Text Hex ?

Decode as ...

Plain

URL

HTML

Base64

ASCII hex

Hex

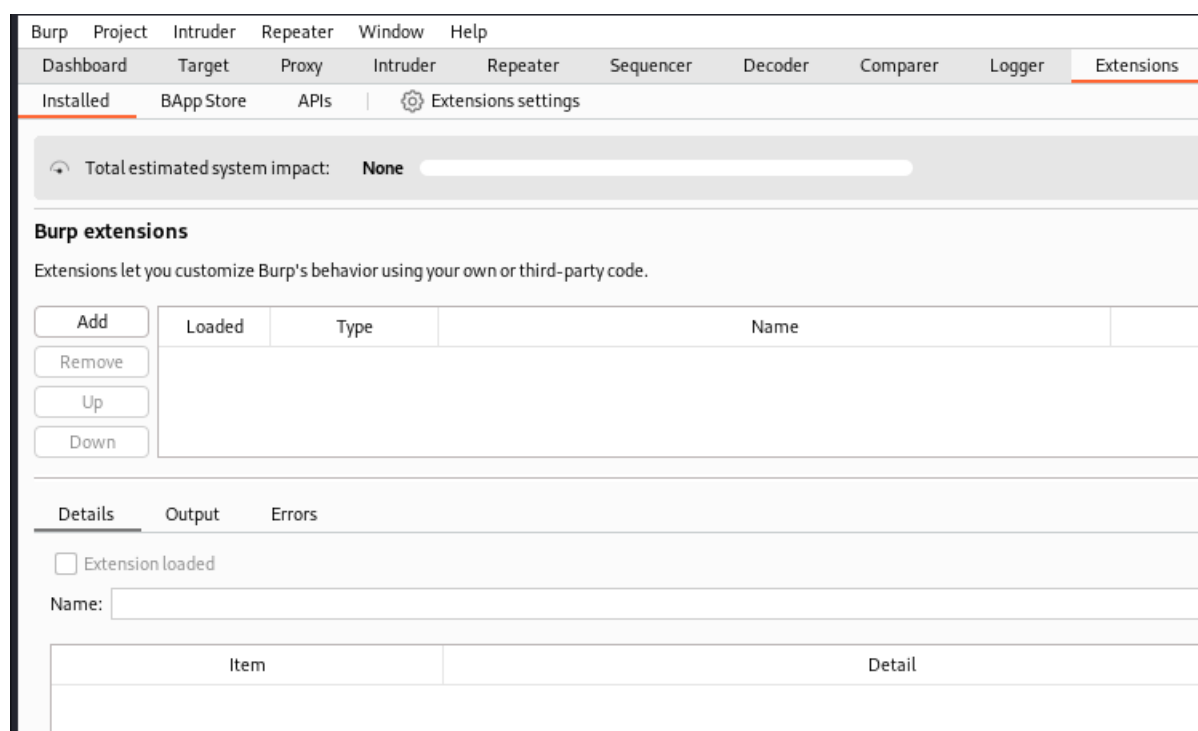
Octal

Binary

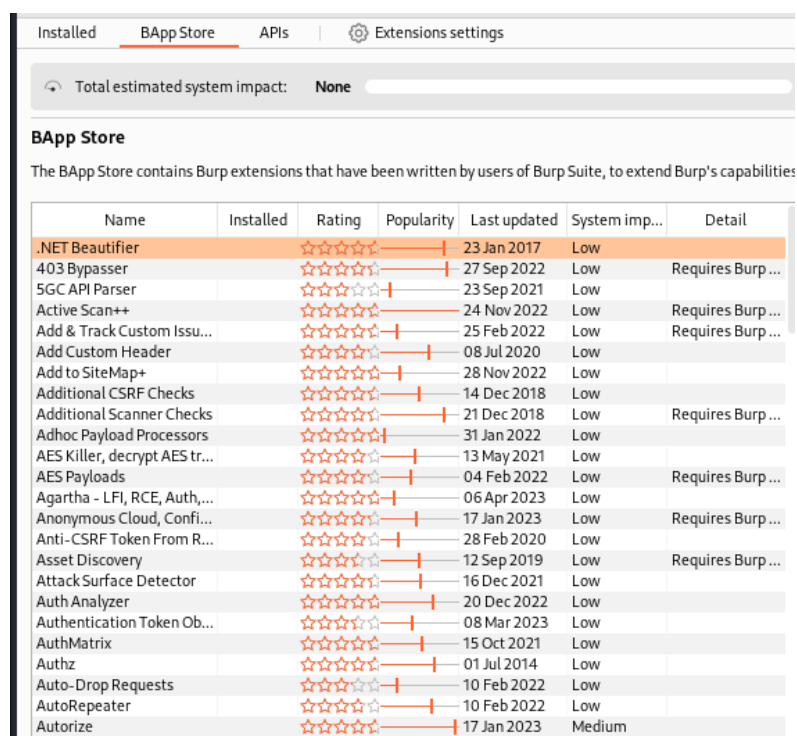
Gzip

Smart decode

Extension is another tool that increase the performance of burpsuite.



In BApp Store we have extra functionality that we can add to our burpsuite installation like a plugin or extensions.



Some of them require burpsuite professional version.

While some are available on free version also.