BURP SUITE

BurpSuite by PortSwigger

- Platform for performing security testing of web applications
- Written in Java. Proprietary and closed source :(
  - 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…

Web Security Academy

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.

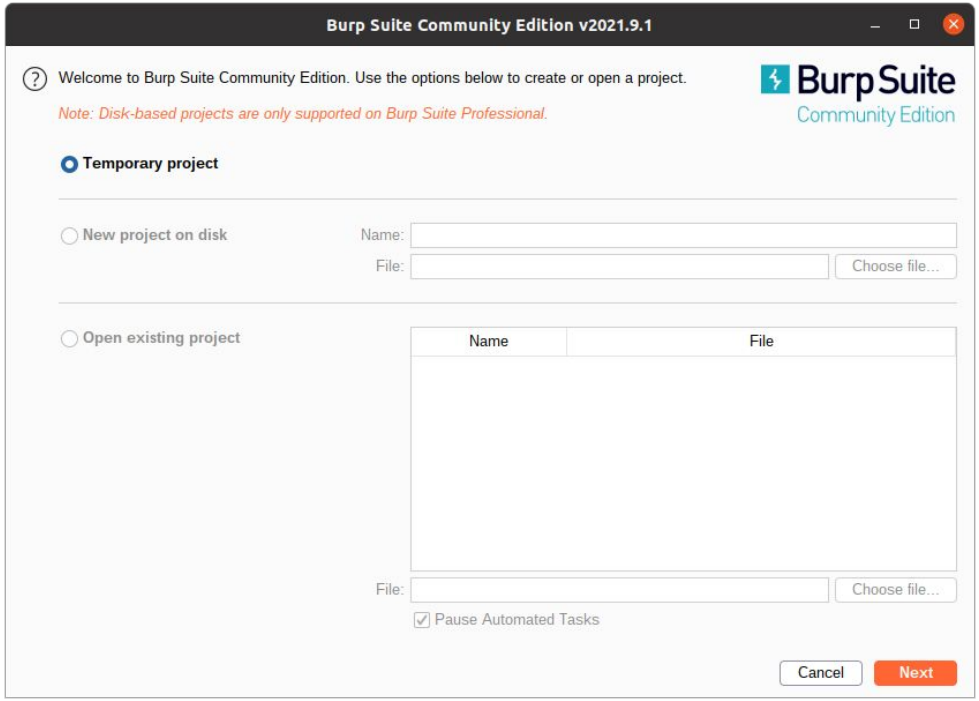
It nicely covers the topics that we will see in the upcoming lectures. The labs may help you prepare for the CTF.

Reference: https://portswigger.net/web-security

Tutorial - Burp Suite (1)

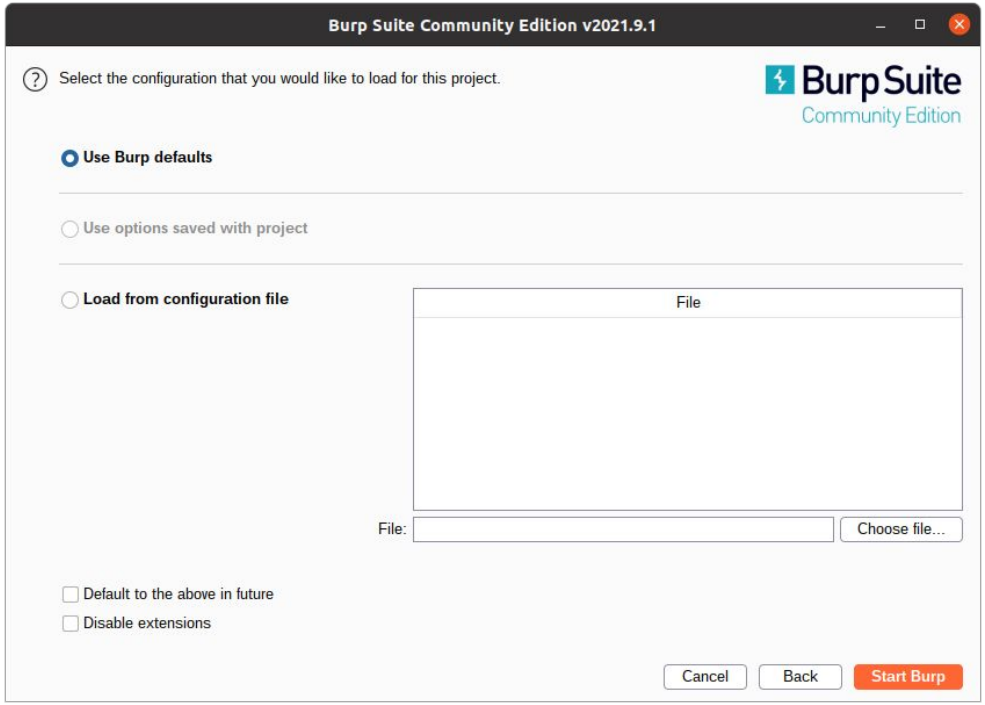
Tutorial - Burp Suite (2)

A temporary project is fine for our goals.



Tutorial - Burp Suite (3)

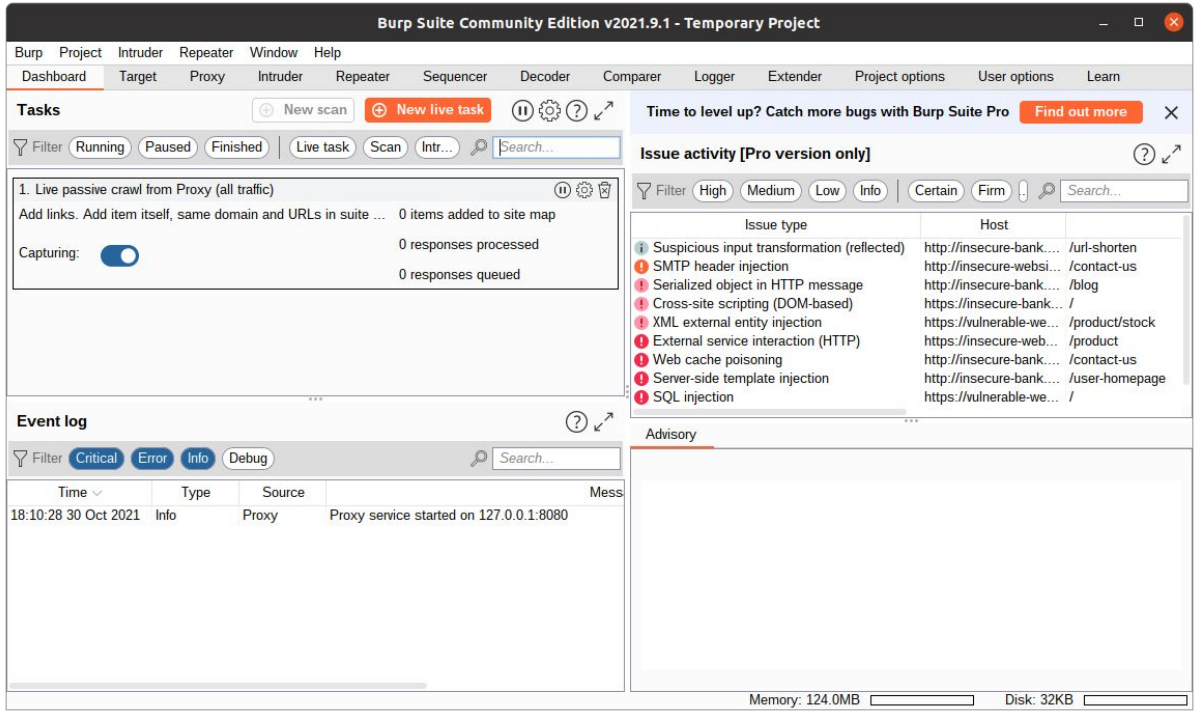
Default settings are fine for our goals.



Tutorial - Burp Suite (4)

Dashboard...

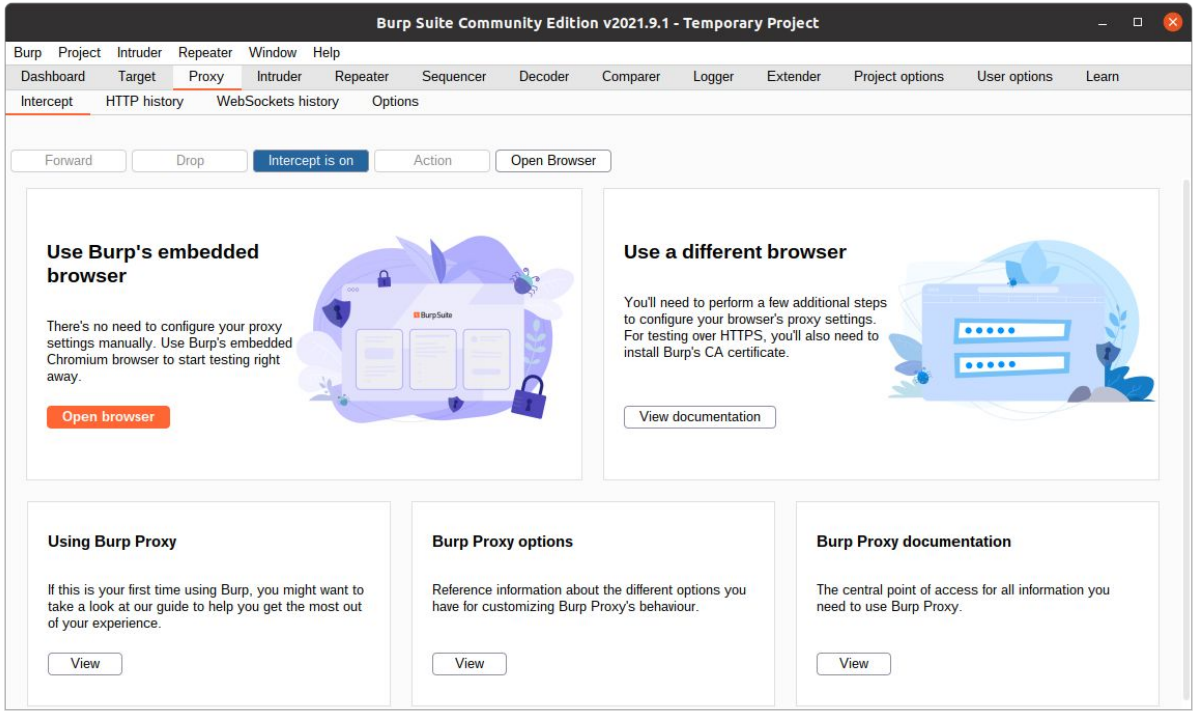
Use the tabs to switch to specific functionalities



Tutorial - Burp Suite (5)

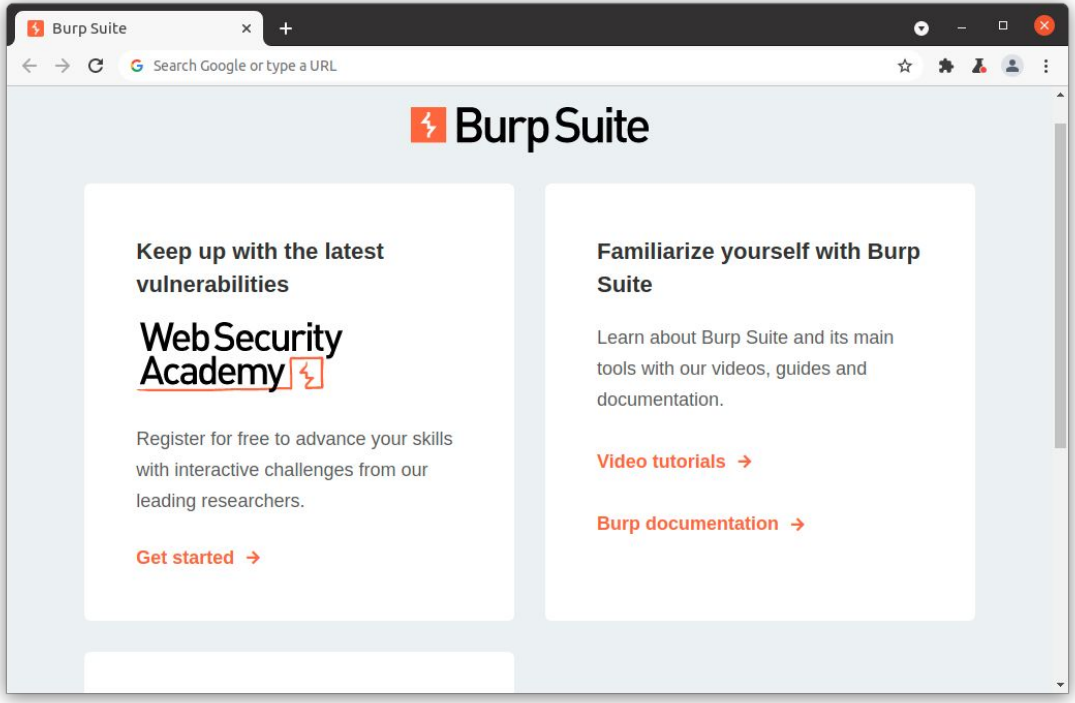
Tab: Proxy > Intercept

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.

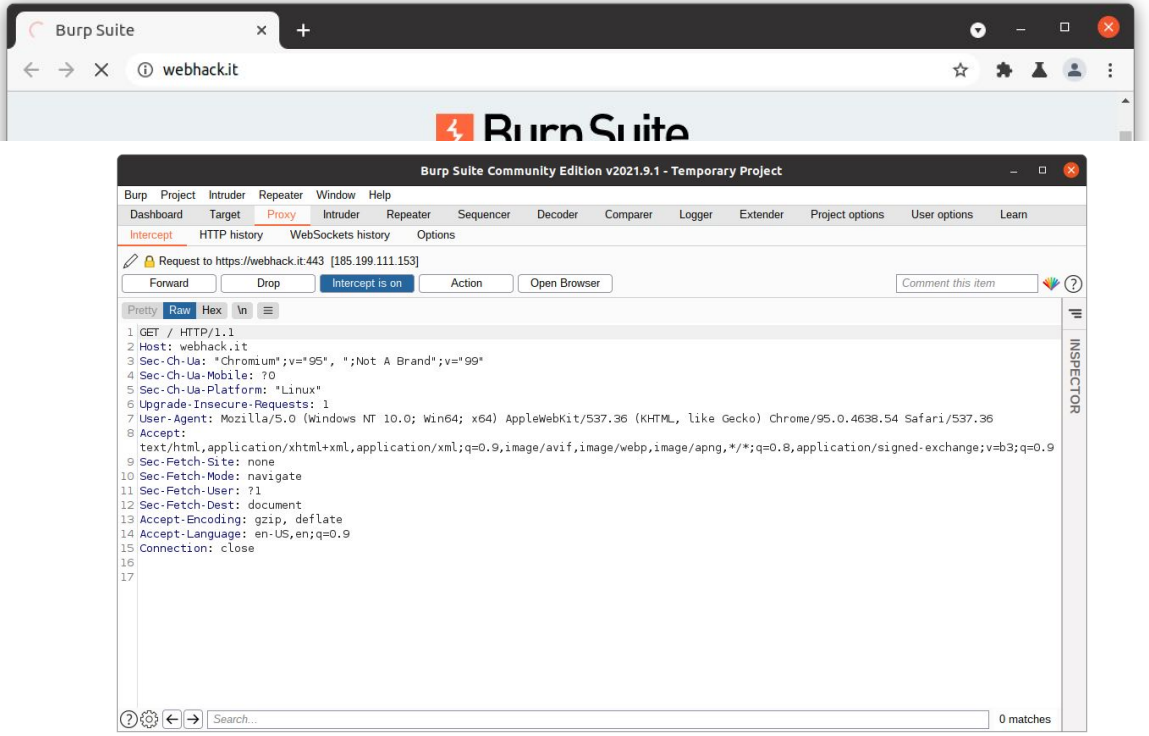


Tutorial - Burp Suite (7)

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

- forward
- drop

You can edit the request before forwarding it.



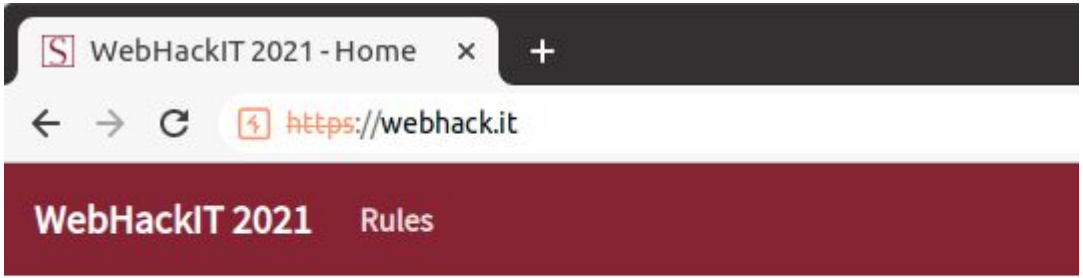
Tutorial - Burp Suite (8)

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)

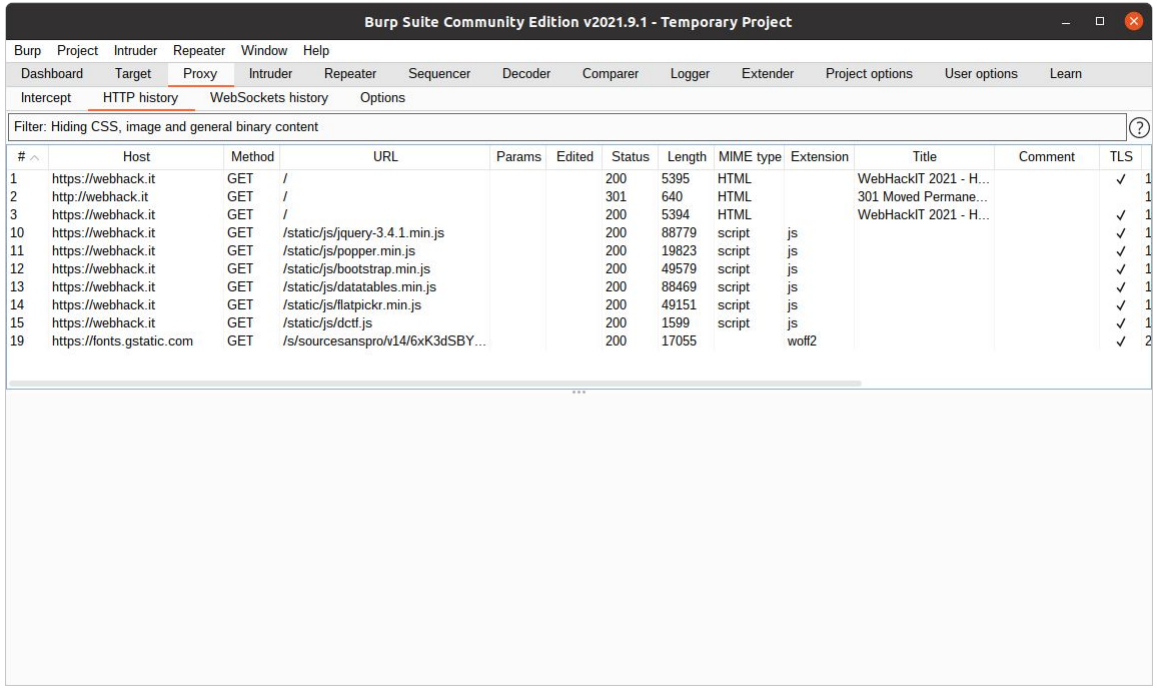
Burp Suite is messing up with the certificates! This is done to correctly intercept our HTTPS traffic.



Tutorial - Burp Suite (9)

Tab: **Proxy > HTTP History**

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

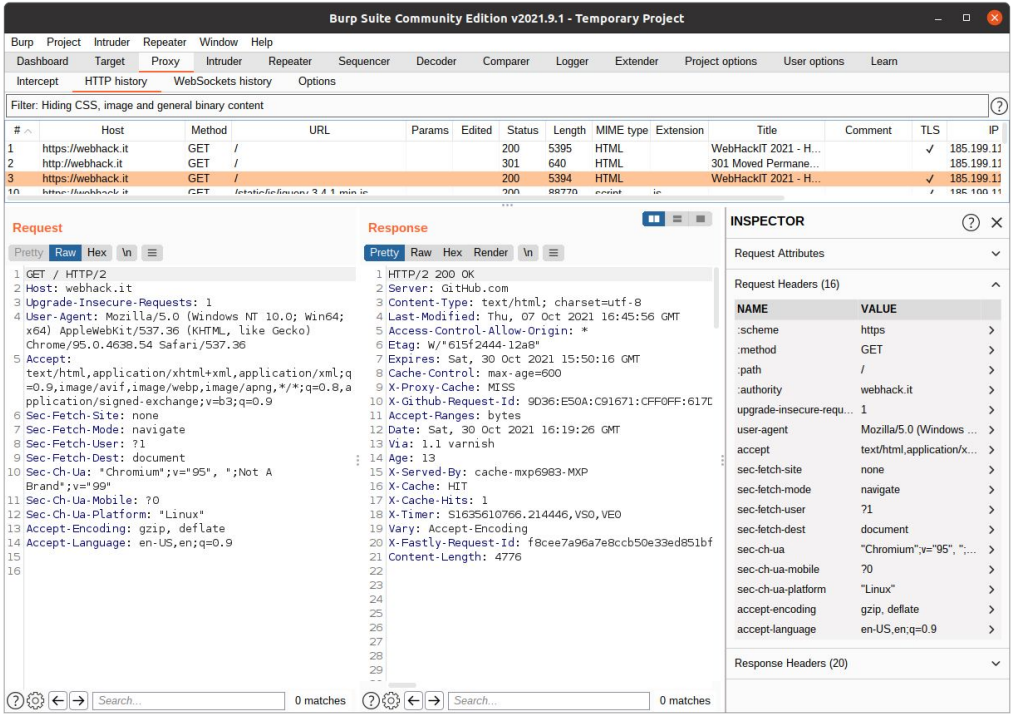


#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	
1	https://webhack.it	GET	/			200	5395	HTML		WebHackIT 2021 - H...		✓	1
2	http://webhack.it	GET	/			301	640	HTML		301 Moved Permane...			1
3	https://webhack.it	GET	/			200	5394	HTML		WebHackIT 2021 - H...		✓	1
10	https://webhack.it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

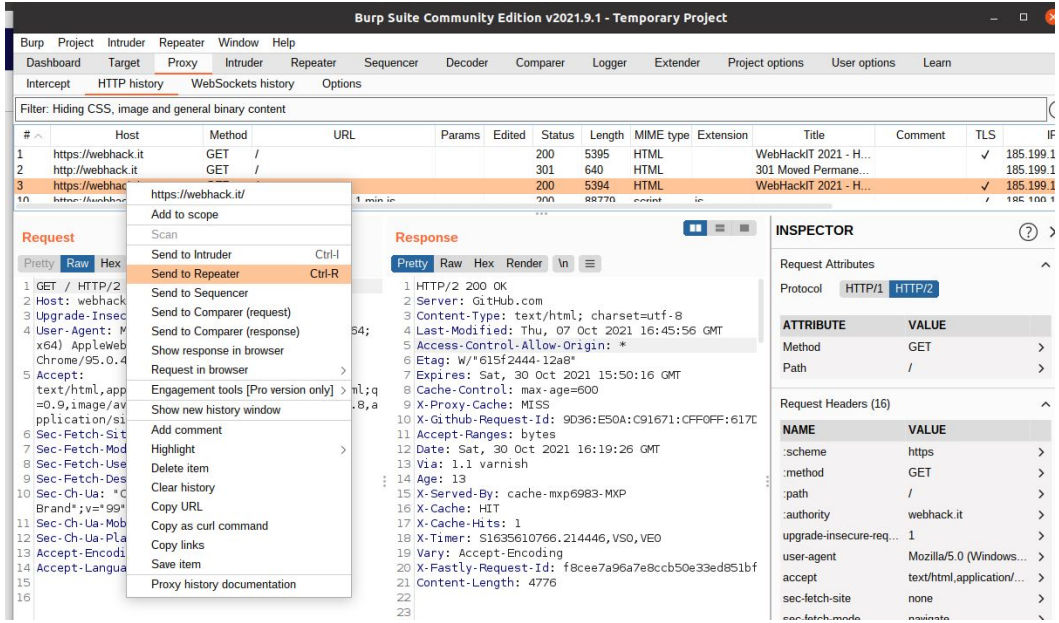
We can see the request and the response.



Tutorial - Burp Suite (11)

Tab: **Proxy > HTTP History**

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



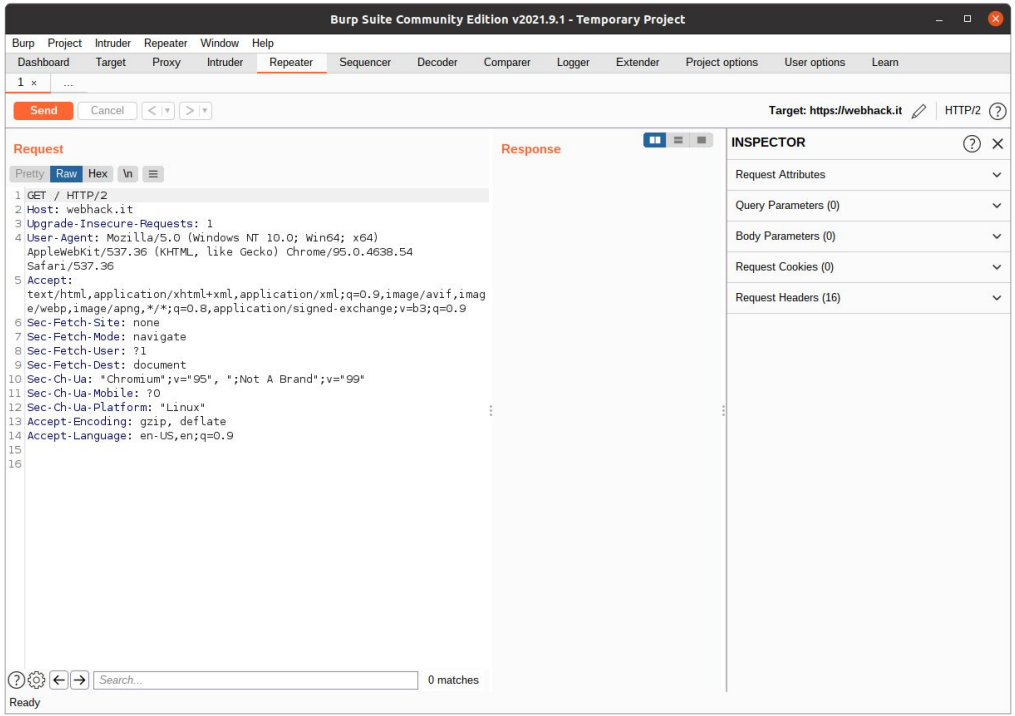
Tutorial - Burp Suite (12)

Tab: Repeater

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

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

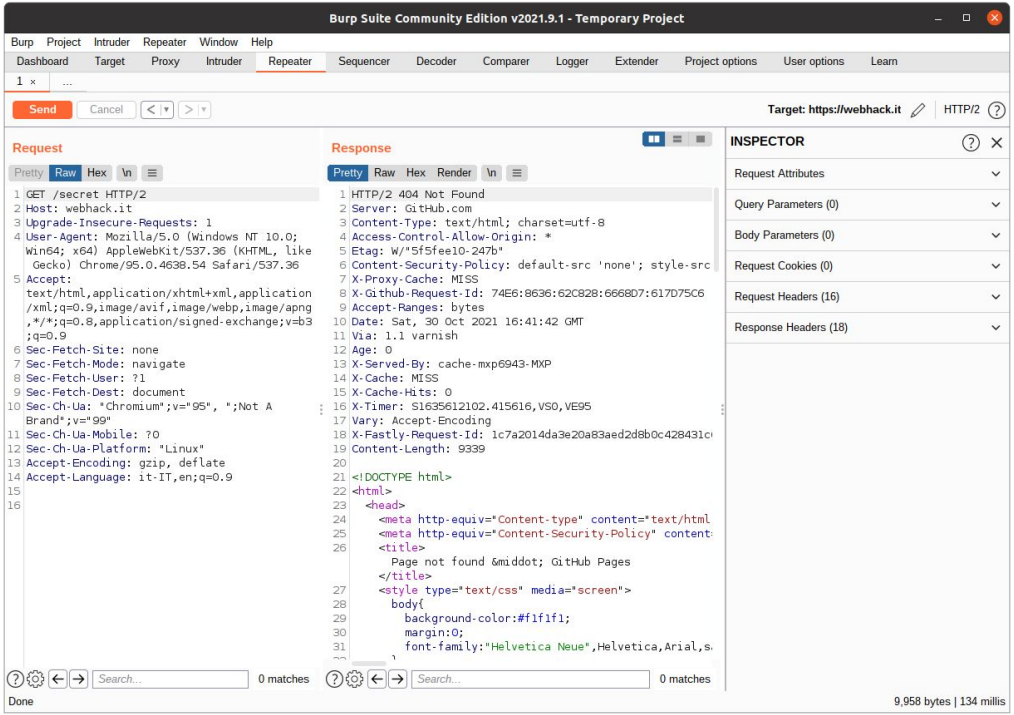
After we can *Send* it.



Tutorial - Burp Suite (13)

Tab: **Repeater**

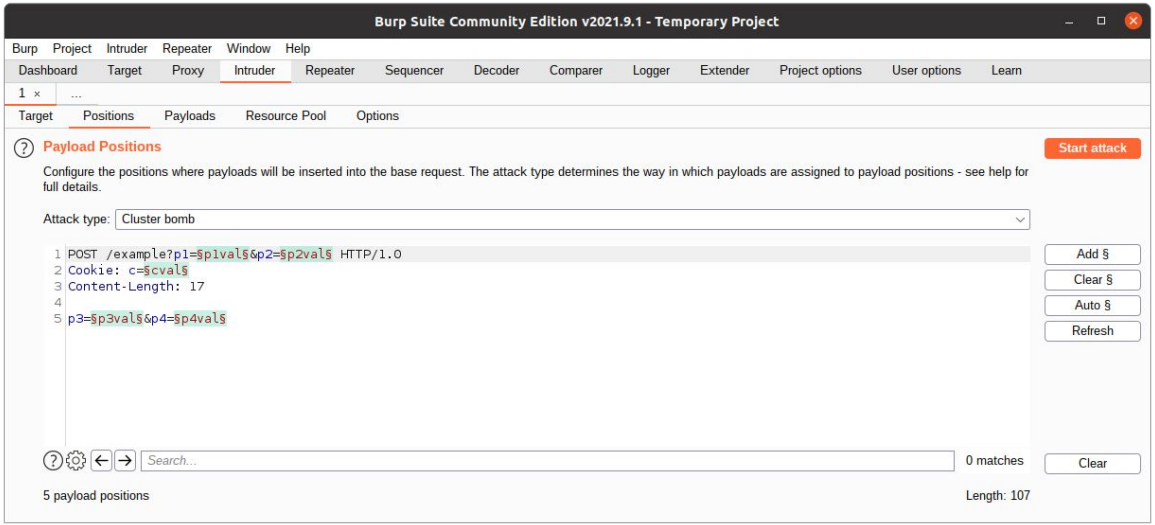
We get back the response. We can use *Render* to view the rendered page.



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).



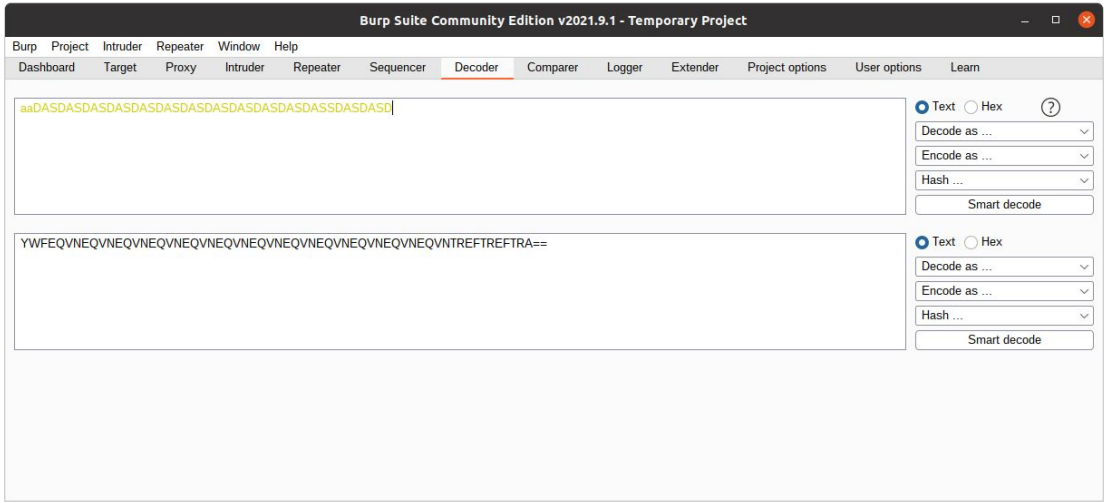
THERE IS NO NEED TO USE THIS FUNCTIONALITY IN OUR CTF

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

username

password

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

```
    .form-signin .form-control {
      position: relative;
      box-sizing: border-box;
      height: auto;
      padding: 10px;
      font-size: 16px;
    }

    .form-signin .form-control:focus {
      z-index: 2;
    }

    .form-signin input[type="email"] {
      margin-bottom: -1px;
      border-bottom-right-radius: 0;
      border-bottom-left-radius: 0;
    }

    .form-signin input[type="password"] {
      margin-bottom: 10px;
      border-top-left-radius: 0;
      border-top-right-radius: 0;
    }
  </style>

</head>

<body class="text-center">
  <form class="form-signin" method="post">
    <h1>Byte Information Exchange</h1>
    <img style="width: 600px;" class="mb-4" src="
    <br />
    <div>
      <!-- debug mmode credentials: demo:base64(demo) -->
      <input type="text" class="form-control" id="user" name="user" placeholder="username">
      <input type="password" class="form-control" id="pass" name="pass" placeholder="password">
      <!--
      <input type="hidden" class="form-control" id="debug_mode" name="debug_mode" placeholder="1" value="0">
      -->
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Enter</button>
    <p class="mt-5 mb-3 text-muted">WebHackIT</p>
  </form>
</body>

</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

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

1 × ...

Send Cancel < > Target: https://training13.webhack.it HTTP/2

Request

Pretty Raw Hex \n

```
1  POST / HTTP/2
2  Host: training13.webhack.it
3  Cookie: challenge_auth_token=
   eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlcmNvcHBhQGd
   tYWlsLmNvbSIsImV4cCI6MTYzNTYyMDQ1NS44NTA3Mjk3LCJpYXQiOjE2MzU
   2MTY4NTUuODUwNzI5N30.Ky3WBBbA3kz6_KZniIVAsznNzdDh57jHHw7f9N8
   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
```

Search... 0 matches

Response

Pretty Raw Hex Render \n

```
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      </style>
80
81    </head>
82
83    <body class="text-center
84      <form class="form-sign
85        <h1>
            <a href="https://w
          </h1>
86        <iframe width="560"
          </iframe>
87        <h1 class="h3 mb-3 f
            Result
          </h1>
88        <textarea class="for
            FLAG: WIT{uaTurfG5
          </textarea>
89
90      </form>
91    </body>
92
```

Search... 0 matches

INSPECTOR

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

Done 2,533 bytes | 14 millis