Sri Lanka Institute of Information Technology.

PicoCTF _Homework

Web Security – [redacted]

Group – [redacted](Y2S2 Cyber Security)

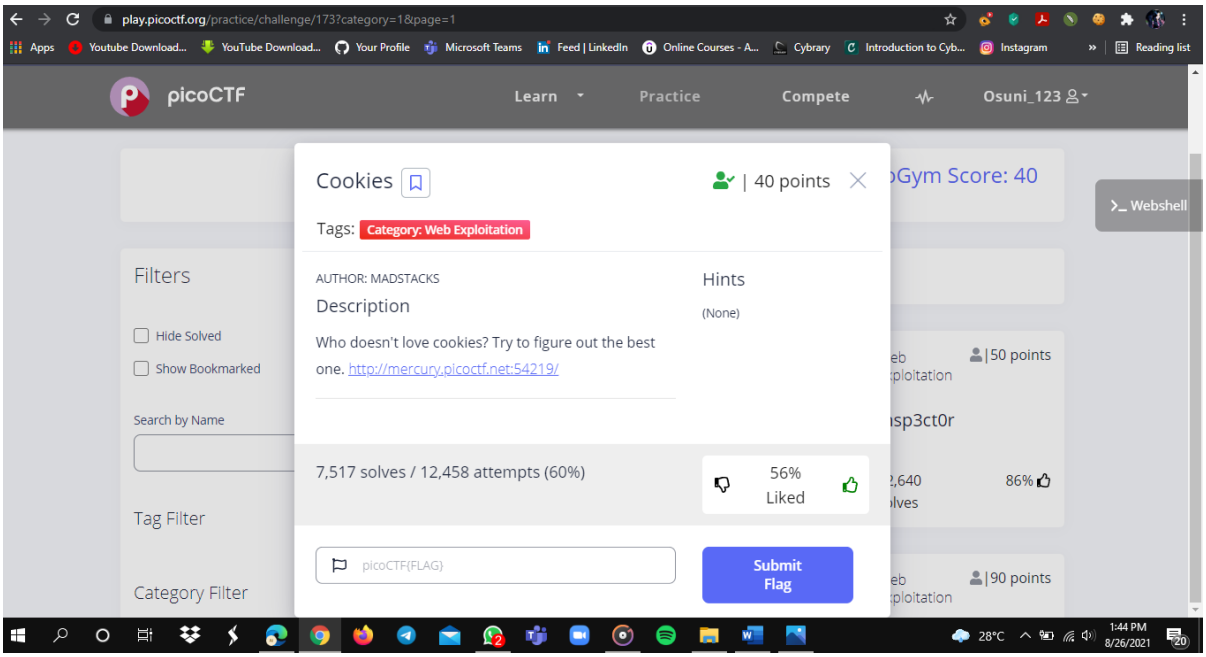
Student Details :

Student ID	Student Name
[redacted]	**Abeywickrama O.D.**

B.Sc. (Hons) in Information Technology

specializing Cyber Security.

Today, I'm going to discuss a CTF that can be found on the picoCTF website, as well as its name. In the web exploitation area, the game I'm going to play today is called "cookies," so let's get started and see how we can tackle this CTF challenge.

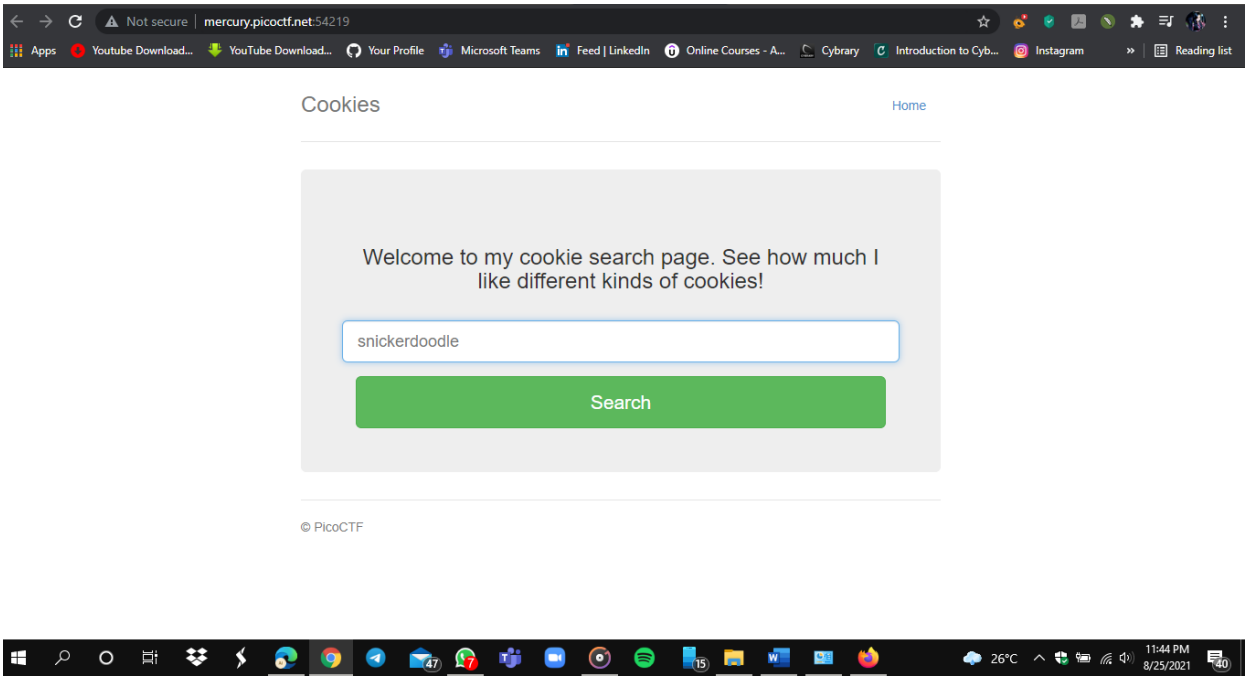


This is the screen that appears when we first click on the cookies CTF, and we can see that there is no suggestion.

They then provided us with a link to the website where we must complete the CTF challenge.

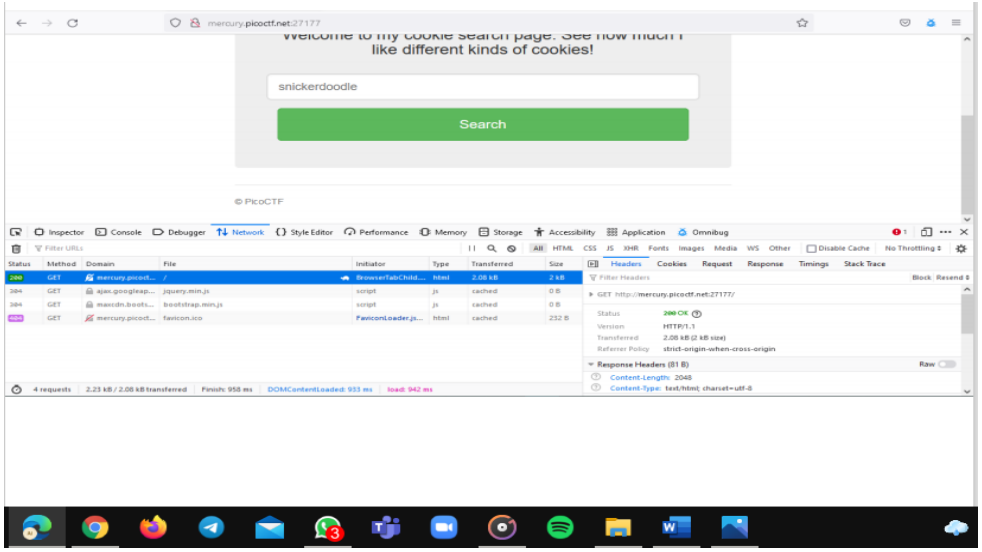
I will insert it over here :- http://mercury.picoctf.net:54219/

When you click the link, you'll be taken to a website that looks somewhat like this.

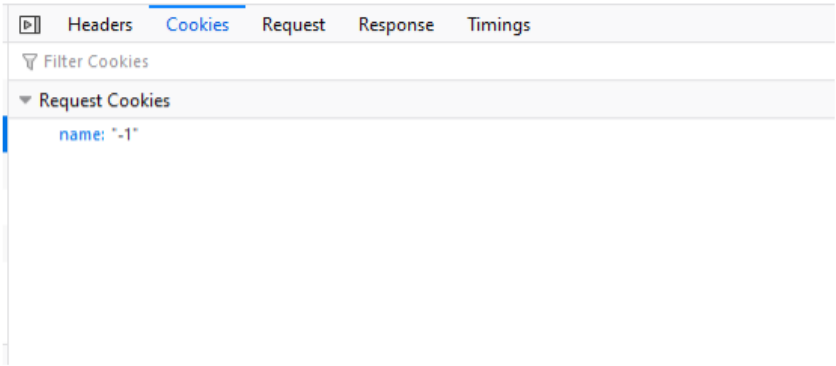


Now we could just browse around and study the webpage to see how it works, so I'll go to the developer settings (right click on the mouse and select developer options) and see what the requests are and what responses were received for the requests that were submitted.

I noticed one response with the status 200 ok among the received responses and decided to investigate it further.

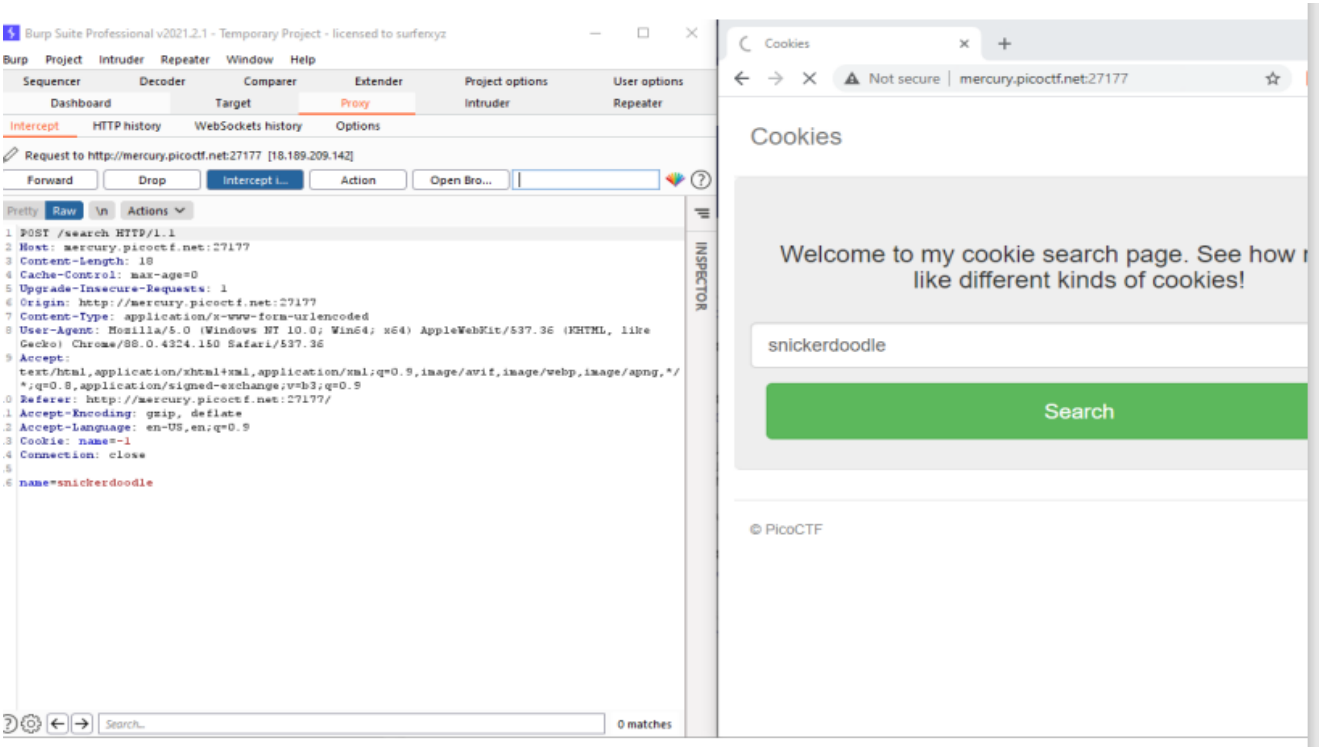


Because this CTF challenge was about cookies, I looked at the cookies section of the 20 ok status response's developer options and discovered that the web cookie's name was set to –“1.”



My web cookie's name was changed from -1 to 0 when I reviewed the response I received.

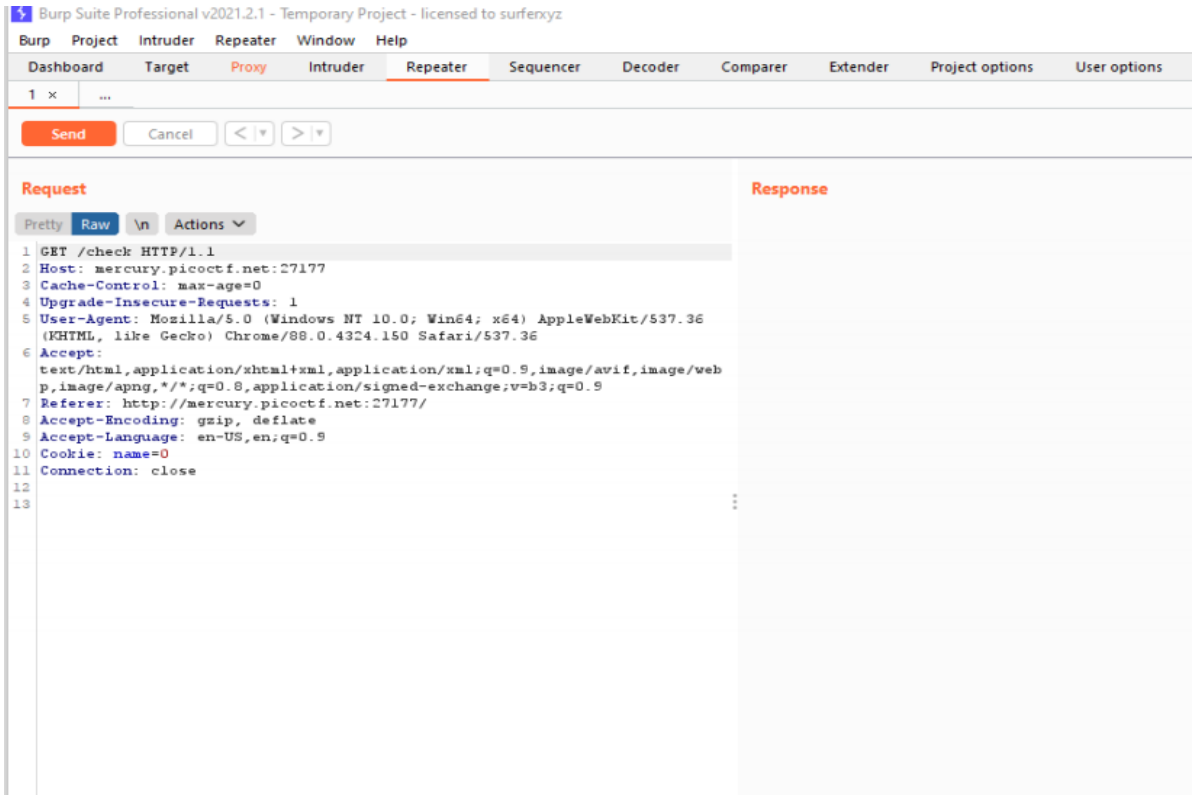
Then, having discovered this, I wanted to see whether it could be brute-forced in some way so that I could capture the flag I was after. As a result, I started bursuite, went into proxy, and activated the burp embedded browser, chrome, and pasted the first link that was supplied to me, searched it, and activated the intercept to see if I could capture any data packets.



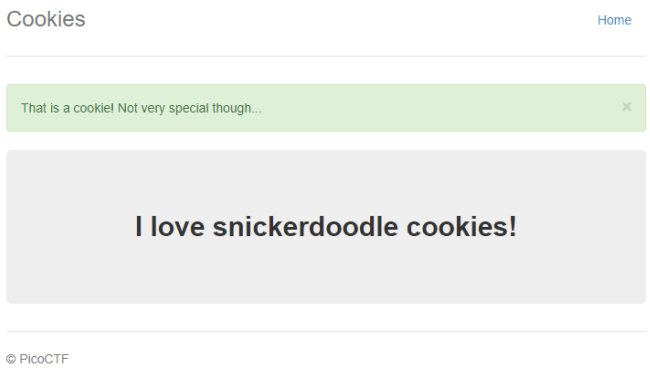
This is what I captured, and the cookie name was set to -1, just like in the developer options, so I figured that if I just sent the request, I'd get 0 as the cookie name. The cookie name changed from -1 to 0 after I routed the request just as I expected.

Then I wondered if I could get an output or find the flag, I was looking for by modifying the cookie name (incrementing the cookie name by 1).

So, I sent this packet to the repeater to accomplish this (by simply right clicking on the mouse and clicking send to repeater)



Once I was in the repeater, I clicked the send button without changing anything to see if I could get a response from the server, and yes, I did get an HTML code in the response, and after a thorough search of the code, I discovered the output I love snickerdoodle cookies, which was similar to the message displayed on the webpage if I had just entered snickerdoodle in the search tab.



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

1 × ...

Send | Cancel

Request

Pretty | Raw | \n | Actions

```
1  GET /check HTTP/1.1
2  Host: mercury.picoctf.net:27177
3  Cache-Control: max-age=0
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
6  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
   p,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7  Referer: http://mercury.picoctf.net:27177/
8  Accept-Encoding: gzip, deflate
9  Accept-Language: en-US,en;q=0.9
10 Cookie: name=0
11 Connection: close
12
13
```

Response

Pretty | Raw | Render | \n | Actions

```
        </button>
40      <!-- <strong>Title</strong> --> That is a cook
41    </div>
42
43
44
45    <div class="jumbotron">
46      <p class="lead">
        </p>
47      <p style="text-align:center; font-size:30px;">
          <b>
            I love snickerdoodle cookies!
          </b>
        </p>
48    </div>
49
50
51    <footer class="footer">
52      <p>
          &copy; PicoCTF
        </p>
53    </footer>
54
55  </div>
56  <script>
57    $(document).ready(function(){
58      $(".close").click(function(){
59        $("myAlert").alert("close");
60      }
        );
61    }
      );
62    </script>
```

Then I wondered whether I could obtain an output if I changed the cookie name from 0 to 1, so I tried it.

Send | Cancel

Request

Pretty | Raw | \n | Actions

```
1  GET /check HTTP/1.1
2  Host: mercury.picoctf.net:27177
3  Cache-Control: max-age=0
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
6  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
   p,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7  Referer: http://mercury.picoctf.net:27177/
8  Accept-Encoding: gzip, deflate
9  Accept-Language: en-US,en;q=0.9
10 Cookie: name=1
11 Connection: close
12
13
```

Response

Pretty | Raw | Render | \n | Actions

```
35      <!-- Categories: success (green), info (b
36
37
38    <div class="alert alert-success alert-dis
39      <button type="button" class="close" dat
          <span aria-hidden="true">&times;</spa
        </button>
40      <!-- <strong>Title</strong> --> That is
41    </div>
42
43
44
45    <div class="jumbotron">
46      <p class="lead">
        </p>
47      <p style="text-align:center; font-size:
          <b>
            I love chocolate chip cookies!
          </b>
        </p>
48    </div>
49
50
51    <footer class="footer">
52      <p>
          &copy; PicoCTF
        </p>
53    </footer>
54
55  </div>
56  <script>
57    $(document).ready(function(){
58      $(".close").click(function(){
```

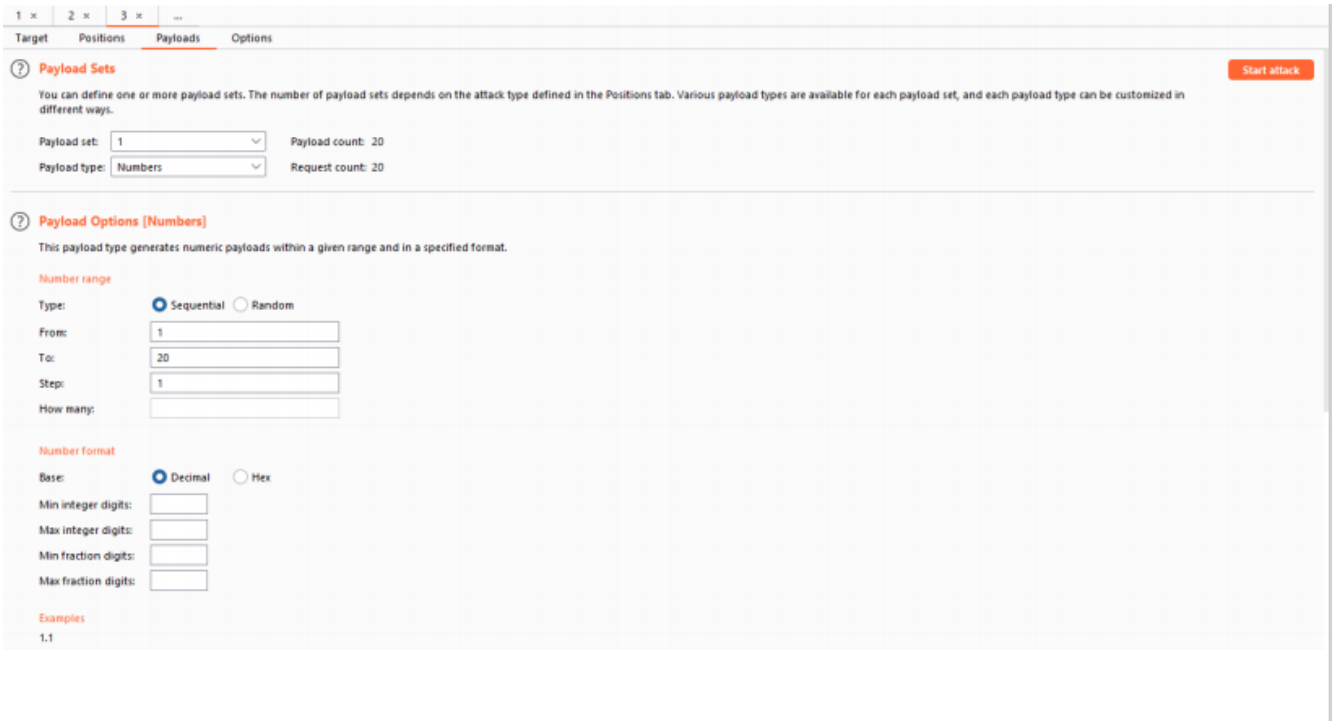
So, since I got a different result this time, I reasoned that if the cookie name was incremented one by one, I'd eventually catch the CTF flag I'm chasing.

Because we are altering the value of the cookie name and capturing the response packets one by one to see the output that we are getting, this is known as bruteforcing, and because it is done manually in the repeater, it is known as manual bruteforce.

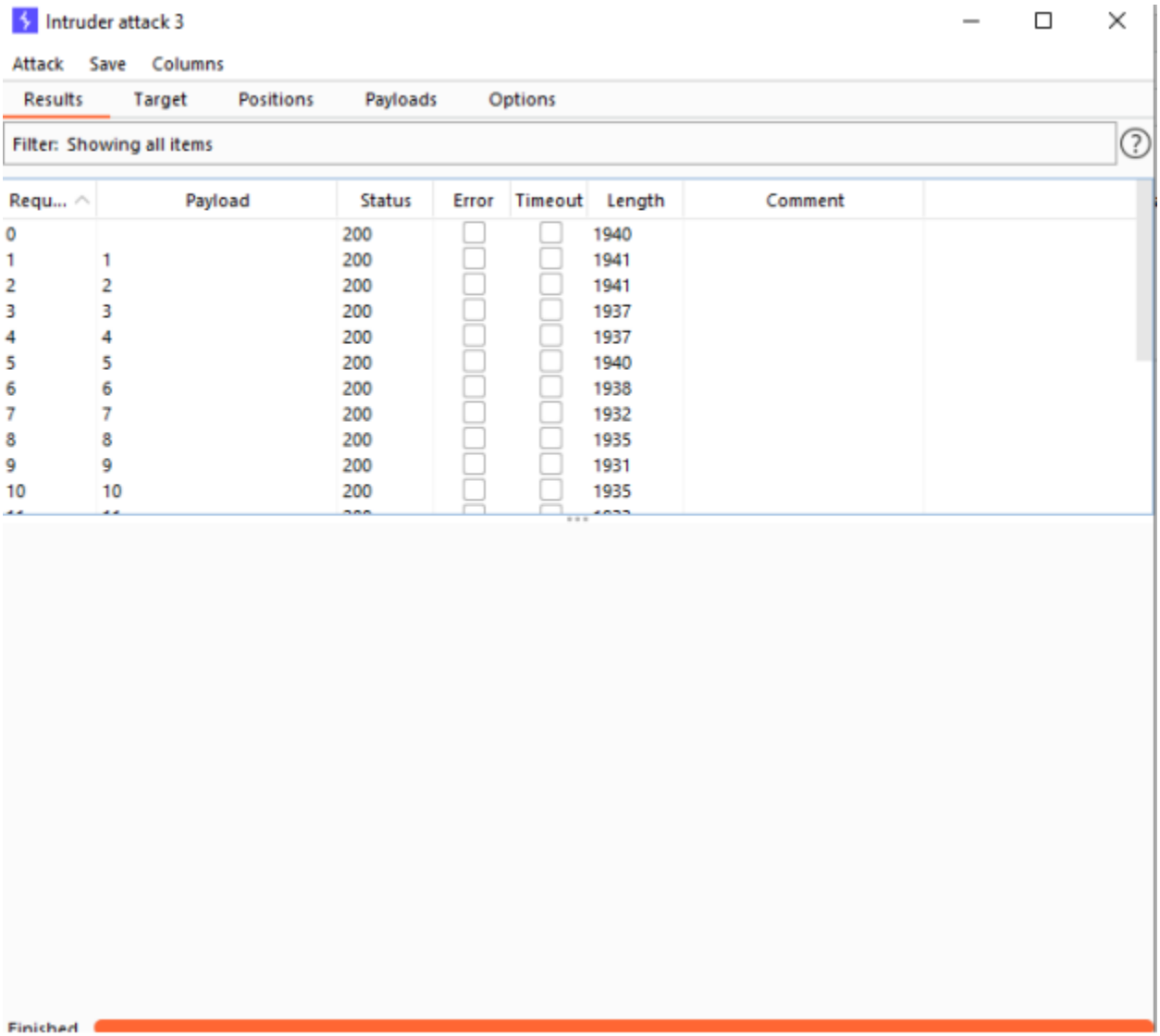
Now that we know it's a bruteforce, but we're not sure how much more we should increment in order to capture the flag, I figured it'd be easier to automate it.

As a result, I returned to the proxy and right-clicked and sent it to the intruder. In intruder, I went to payloads and changed the payload type to numbers, as well as the payload settings. I intended to bruteforce from 1 to 20.

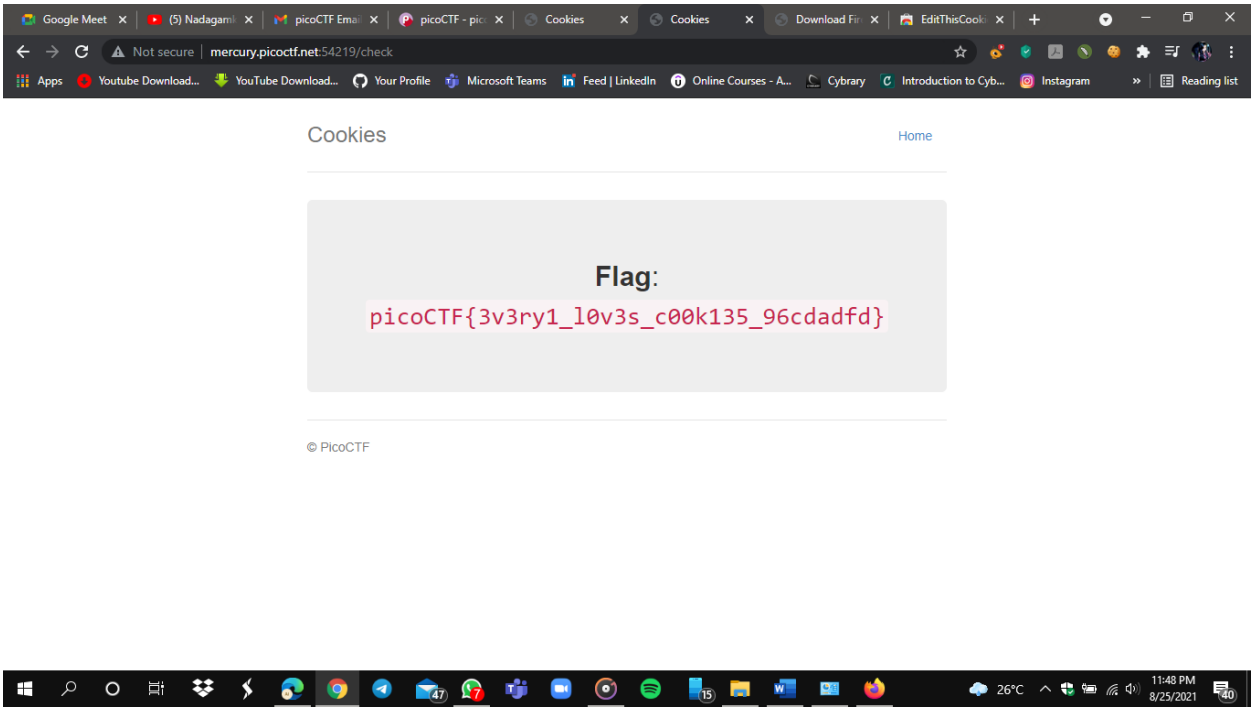
As a result, I added the necessary values to it one step at a time, as seen in the image below.



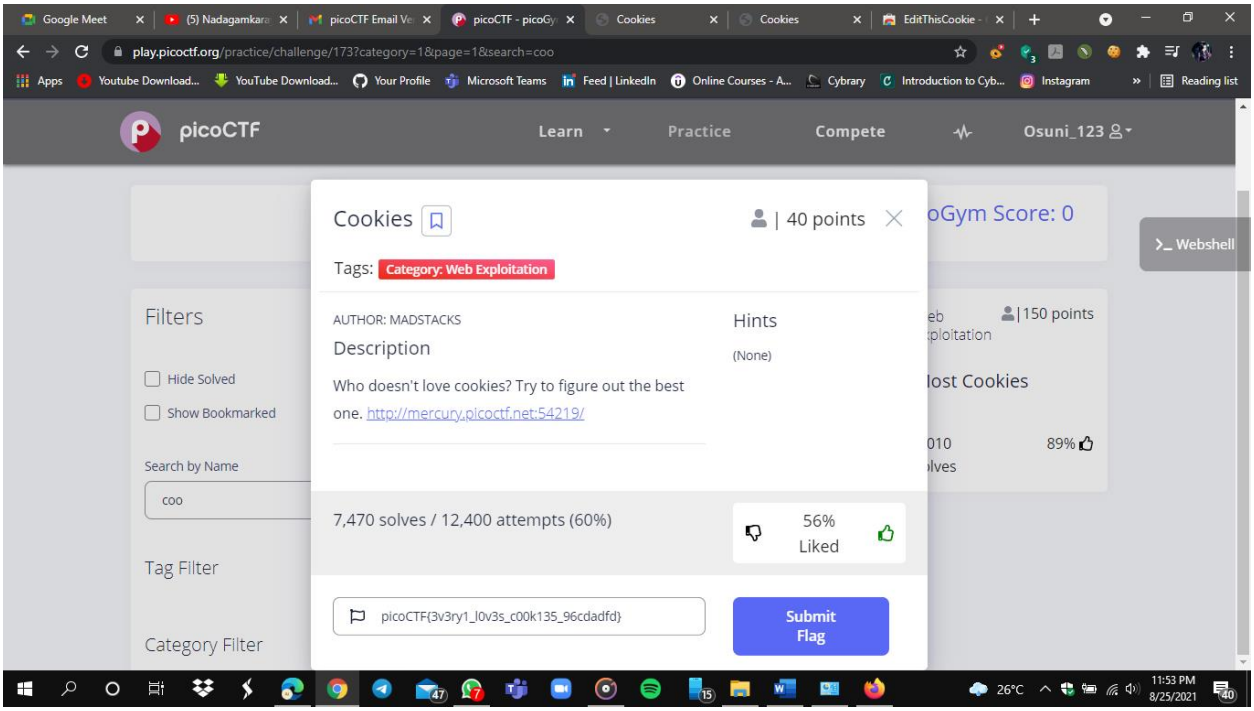
Finally, I pressed the start attack button and waited for the attack to complete before going through the responses one by one.



Finally, after a little more digging, I came upon the 18th request, which had a response that was notably different from the rest of the responses I had received since it contained a code of some sort that looked something like this.



picoCTF{3v3ry1_l0v3s_c00k135_96cdadfd}



It had picoCTF and a code inside these brackets, so I decided to submit it because it looked like the flag, I'd been looking for all along, so I went in and submitted it, and yes, it was the flag I'd been looking for all along.

That concludes today's challenge on the picoCTF website, the Cookies CTF challenge.