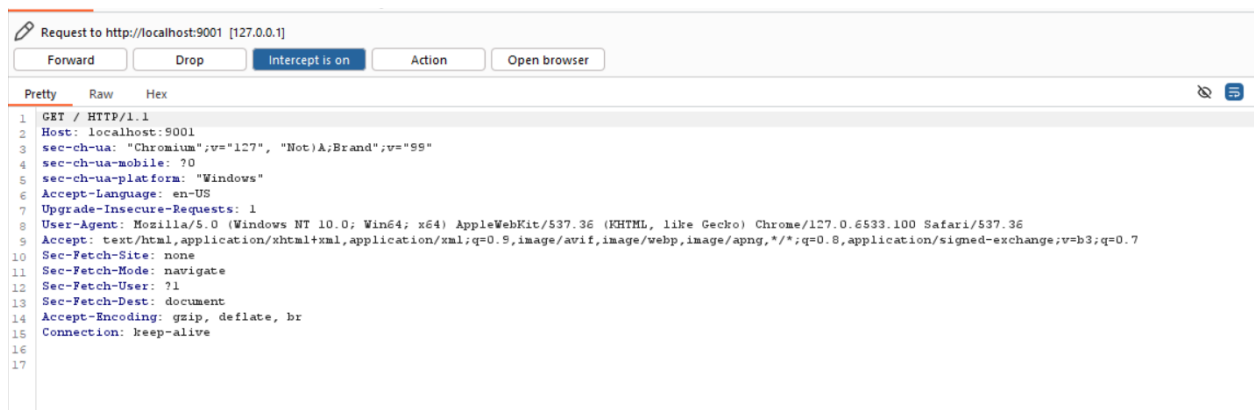


Weather

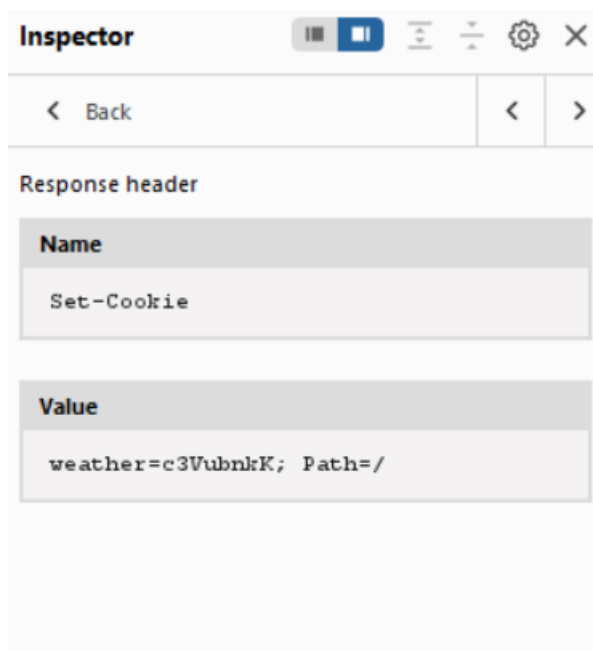
Nom-nom-nom

Well, nom-nom-nom was a giveaway. I knew cookies are used in ctf in one way or another, so I had a direction where I had to look.

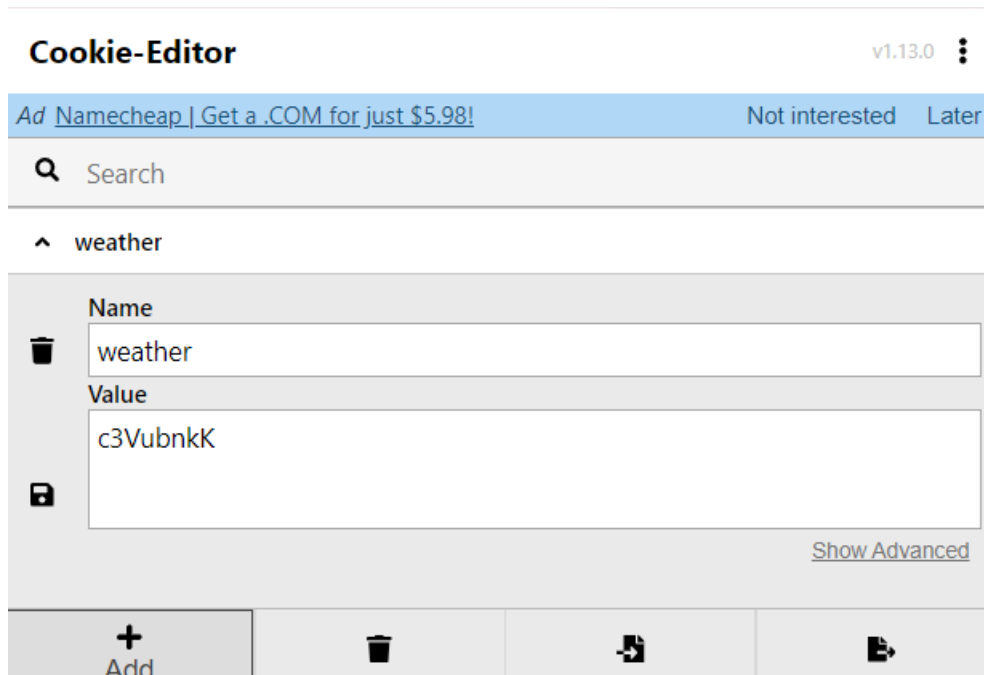
I started using burp suite like all the other challenges, because it was the only proper tool I knew how to partially use.



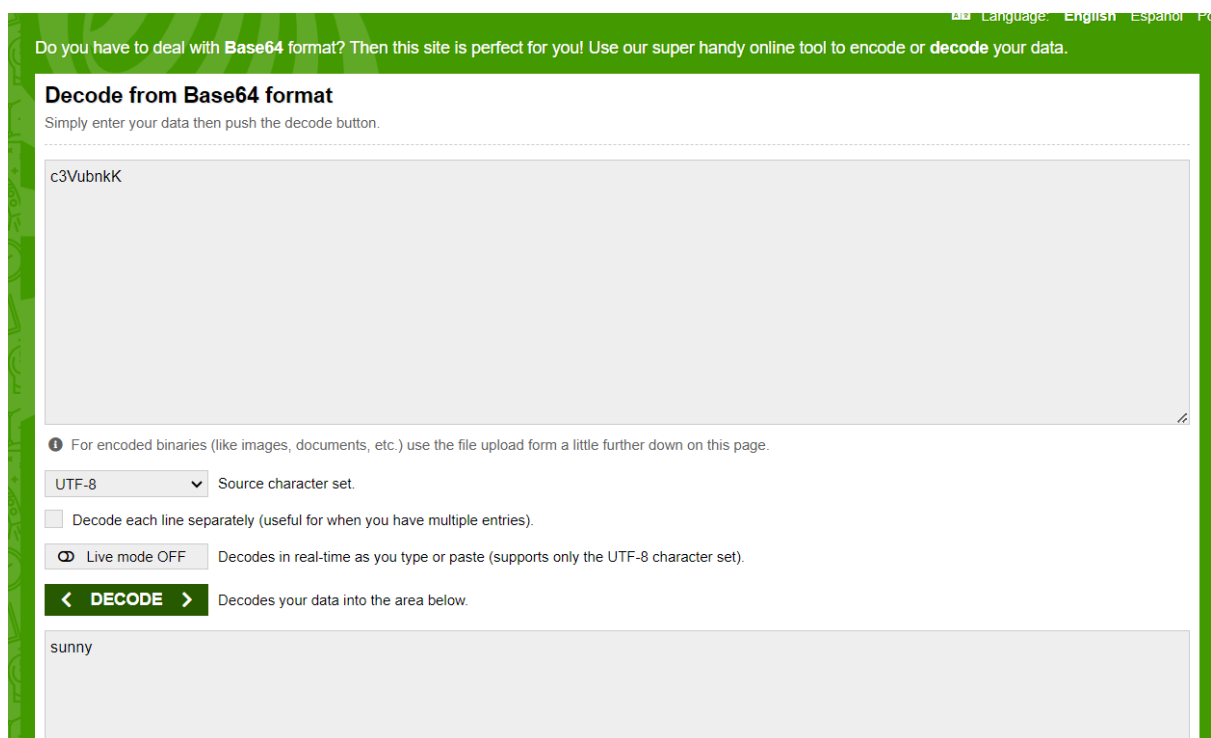
I forwarded the intercepted request, and then dived deeper.



With burpsuite, I got to know that the cookie for the website is “c3VubnkK”.



I used a base64 decoder to decode the value of the cookie, and I got “sunny” as the output.

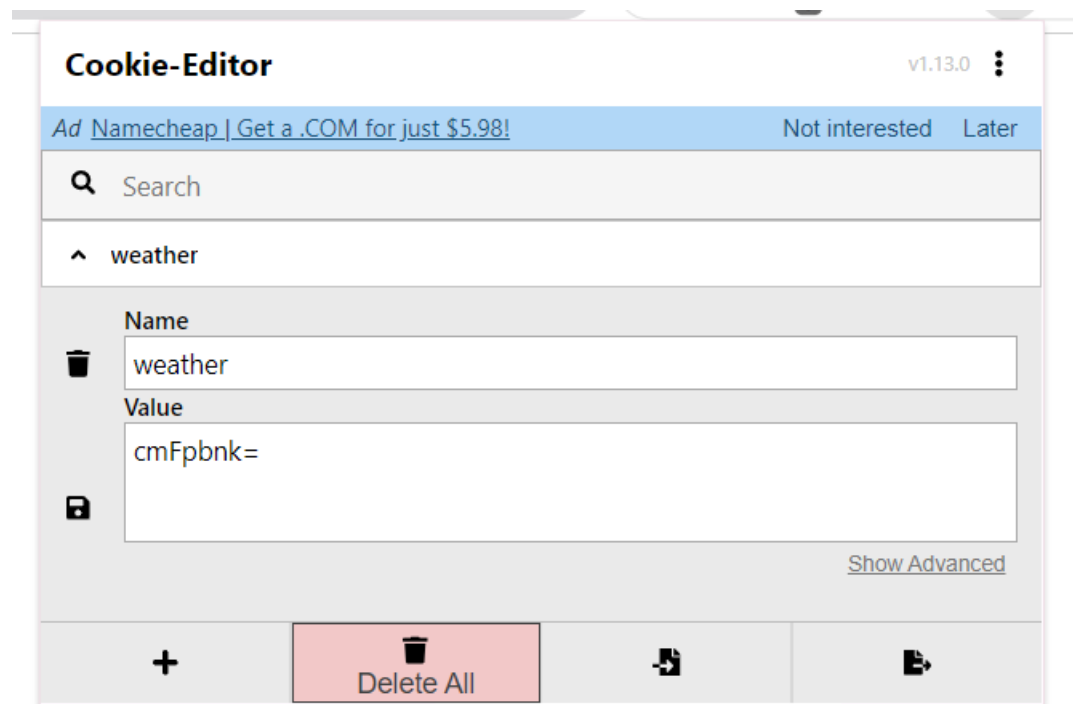


I only like rainy weather ;-;

This is what the website said. So now I had to change the weather from sunny to rainy.

So, I entered rainy in the base64 encoder and then used it as the cookie.

For that I added a cookie editor extension.



Encoding “rainy” gave me this as an output: “cmFpbmk=”

### Encode to Base64 format

Simply enter your data then push the encode button.

rainy

**To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.**

UTF-8

 Destination character set.

LF (Unix)

 Destination newline separator.

☐

 Encode each line separately (useful for when you have multiple entries).

☐

 Split lines into 76 character wide chunks (useful for MIME).

☐

 Perform URL-safe encoding (uses Base64URL format).

☒ Live mode OFF

 Encodes in real-time as you type or paste (supports only the UTF-8 character set).

**> ENCODE <**

 Encodes your data into the area below.

cmFpbmk=

I used it in the cookie editor and the flag was triggered, giving tarush his favourite rainy weather and me my sweet points.

←

→

↻

🔒 localhost:9001

ehax{baarlsh\_n3\_bhlg4\_dly4}