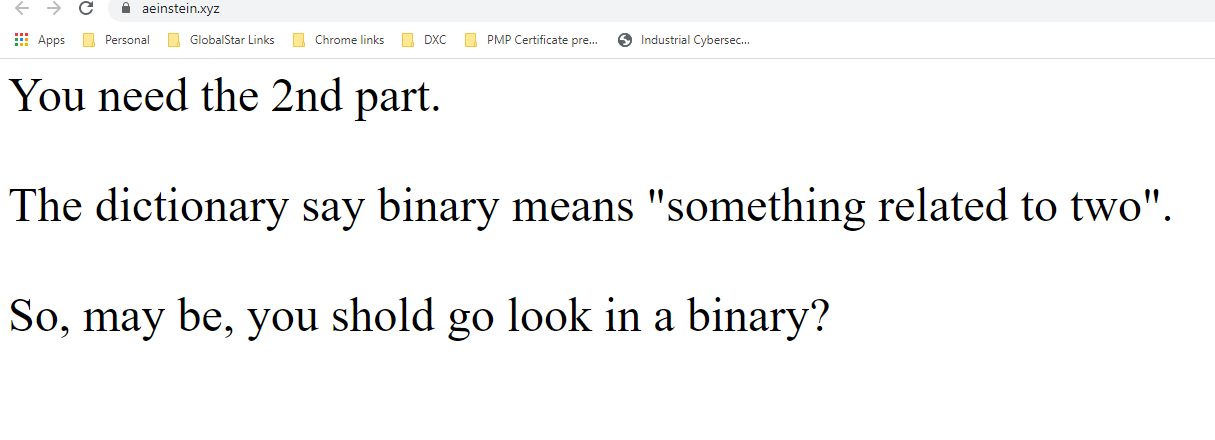
# Question-1

Flag Captured: **4621\_ctf{not\_a11\_sh4rks\_mak3\_y0u\_cry\_s0me\_l3t\_y0u\_spy}**

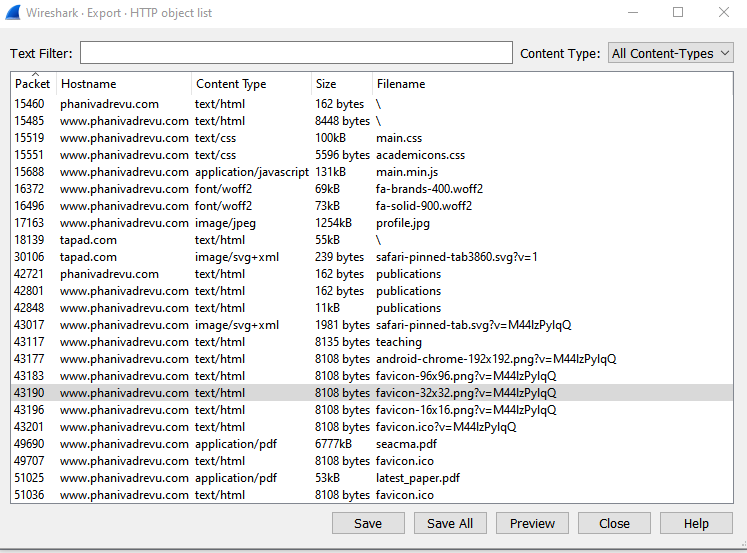
Opened the dump.pcapng file in wireshark. Used Find a packet option in Wireshark and searched for the text ‘ctf’ because we know that the flag format is 4621\_ctf. Using this option I found the packet 9745 in which the first part of the flag is found 1 of 2: 4621\_ctf{not\_a11\_sh4rks\_mak3\_y0u\_

To find the second part of the flag. I further investigated the packet 9745. This packet seems to be a response sent by the url <http://aeinstein.xyz>

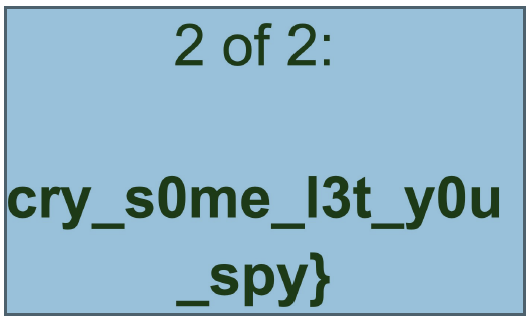
The website showed this message,



Which gives a hint to look in the binary. To export all the files transferred in the network traffic Wireshark has an option File 🡪 Export Objects 🡪 http. This option gave the HTTP Object list.



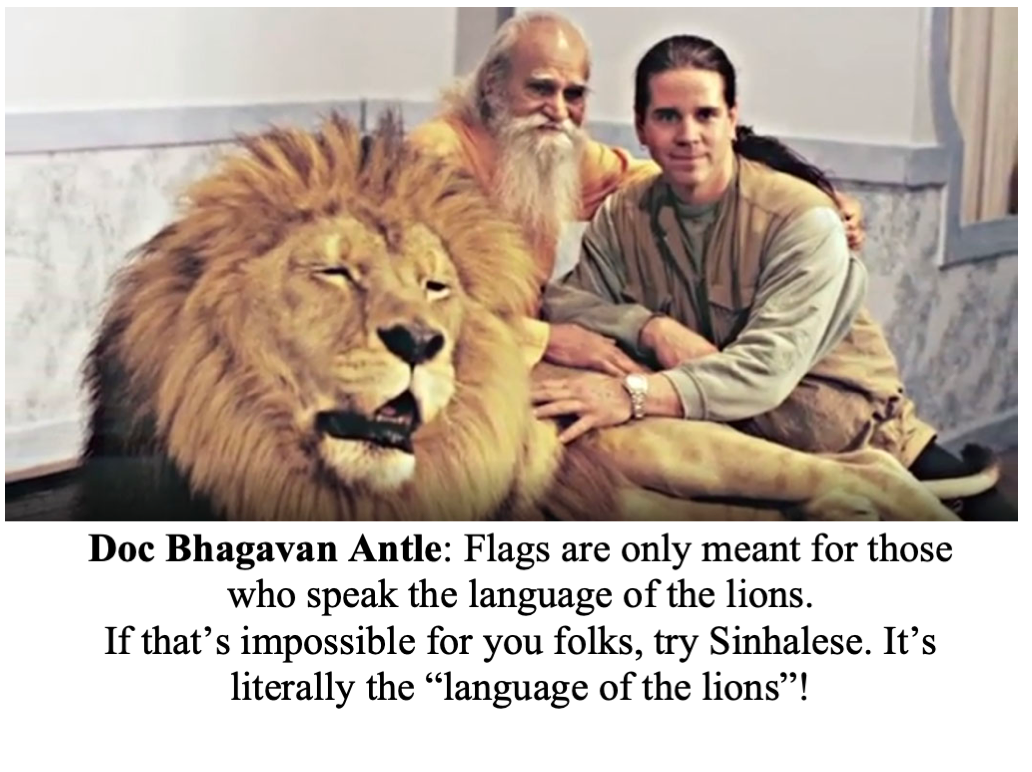
I saved all the Files and was able to find the second part of the flag in the pdf file latest\_paper.pdf



# Question-2

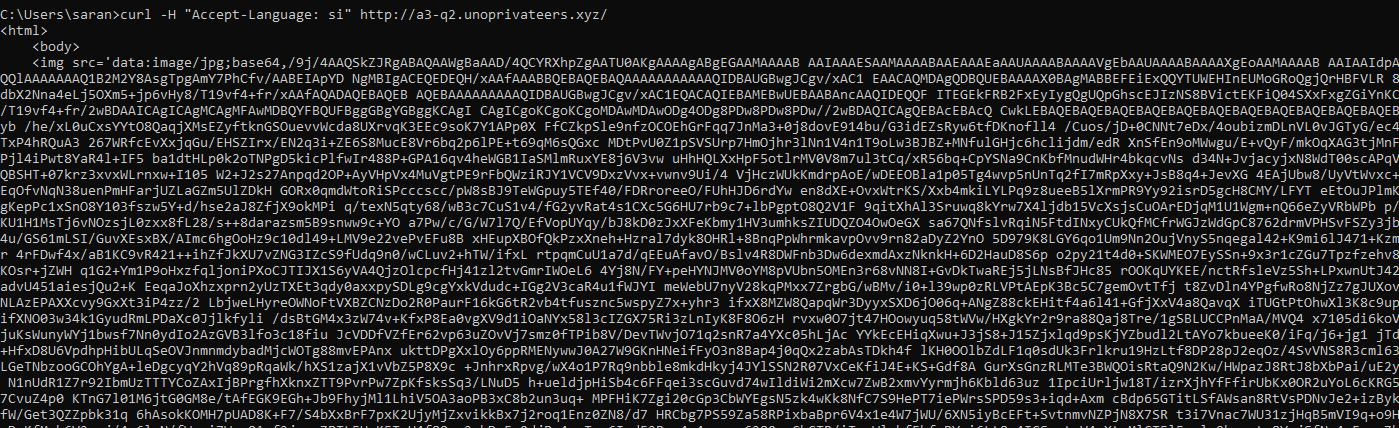
Flag Captured: **4621\_ctf{tiger\_king\_is\_the\_new\_thing}**

This is a website header changing CTF challenge. The website given <http://a3-q2.unoprivateers.xyz/> showed the below picture



This website just had a picture saying speak language of lions which is the language of Sinhalese. So I tried sending a curl request by changing the header Accept-Language: si.

*Curl -H “Accept-Language: si”* [*http://a3-q2.unoprivateers.xyz/*](http://a3-q2.unoprivateers.xyz/)



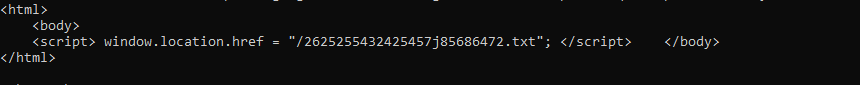
I got an image with Carol Baskin



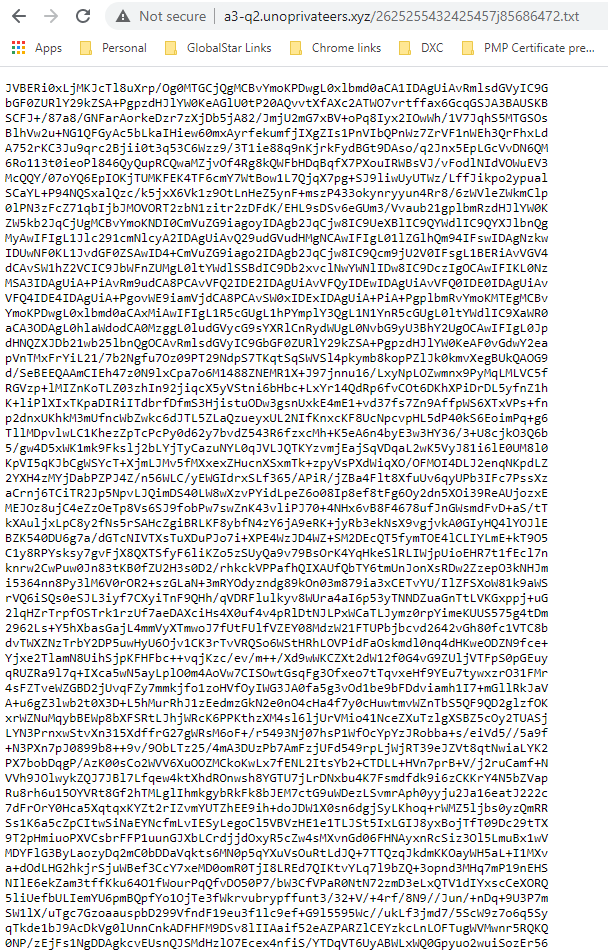
This hint says do not use any normal browser. Which means I need to change the User-Agent header. I tried sending a curl request with Header option FireTiger, FireLion and FireCat.

*curl -H "Accept-Language: si" -A User-Agent:FireCat* [*http://a3-q2.unoprivateers.xyz/*](http://a3-q2.unoprivateers.xyz/)

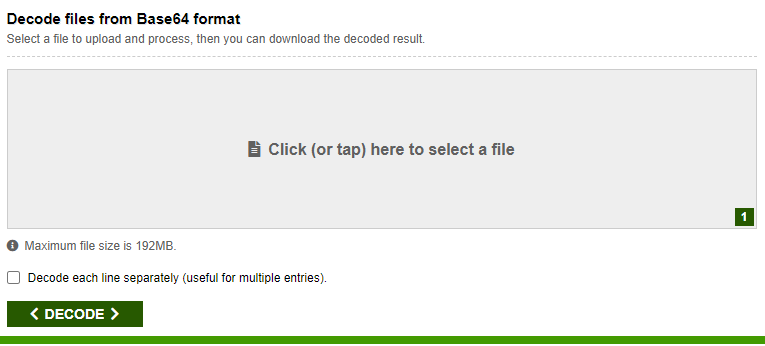
When I used the User-Agent FireCat I got a different response as shown below

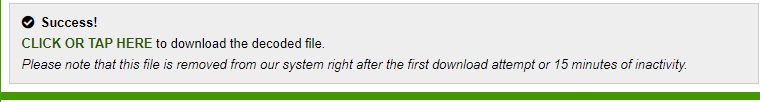


I got a reference to a txt file with some random base64 characters



I saved this text file and used a online base64 decoder <https://www.base64decode.org/> uploaded the file into this website





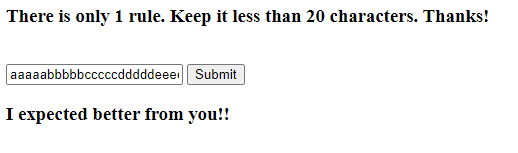
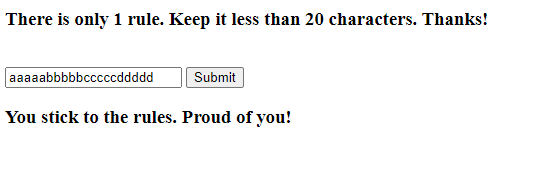
and I got a decoded pdf file which when opened showed me the flag for this challenge



# Question-3

Flag captured: **4621\_ctf{rul3\_breakers\_r\_fl4g\_t4ker5}**

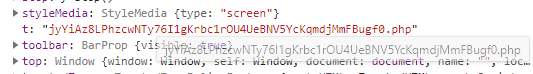
This is a JavaScript task. Using Chrome developer options source tab I was able to see the JavaScript used to render the response text on the page.

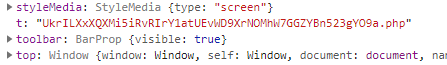


The website renders a h3 element based on the input text’s length. If the input length is less than 20 it displays “*You stick to the rules*” else “*I expect better from you*”

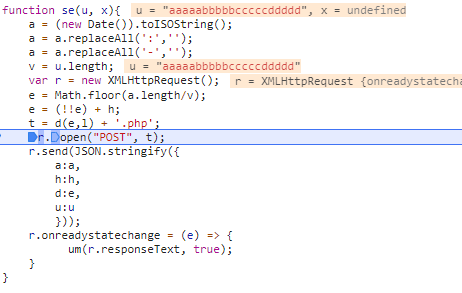
To understand how it is rendering these outputs I started adding breakpoints in the script and started debugging the code. The function se(u,x) seems to be the main function in the script. This function based on the length of the input text it calls a different .php file.

If length less than 20 jyYiAz8LPhzcwNTy76I1gKrbc1rOU4UeBNV5YcKqmdjMmFBugf0.php is called else UkrILXxXQXMi5iRvRIrY1atUEvWD9XrNOMhW7GGZYBn523gYO9a.php is called





I assumed to get the Flag; I should call the jyYiAz8LPhzcwNTy76I1gKrbc1rOU4UeBNV5YcKqmdjMmFBugf0.php service with a proper input parameter.



But if I use the input box in the website I will not be calling the php script that I want always. So to bypass this behavior, I wrote my own function ac\_lt\_func(text) similar to se(u,x) and tried to call the jyYiAz8LPhzcwNTy76I1gKrbc1rOU4UeBNV5YcKqmdjMmFBugf0.php This function which I wrote always calls the php script which runs when input length is less than 20 with the POST parameters a,h,d,u

*function ac\_lt\_func(inp)*

*{*

*a = (new Date()).toISOString();*

*a = a.replaceAll(':','');*

*a = a.replaceAll('-','');*

*//Skipping the part of verifying length*

*h = 'rules.unoprivateers.xyz';*

*// setting e to true*

*e = 'true' + h;*

*u = inp;*

*t = 'jyYiAz8LPhzcwNTy76I1gKrbc1rOU4UeBNV5YcKqmdjMmFBugf0.php';*

*var r = new XMLHttpRequest();*

*r.open("POST", t);*

*r.send(JSON.stringify({*

*a:a,*

*h:h,*

*d:e,*

*u:u*

*}));*

*r.onreadystatechange = (e) => {*

*um(r.responseText, true);*

*}*

*}*

The server is currently returning 502 Gateway error. So, I could not take a screenshot of the working of this JavaScript that I wrote. Luckily, I noted down the Flag earlier

