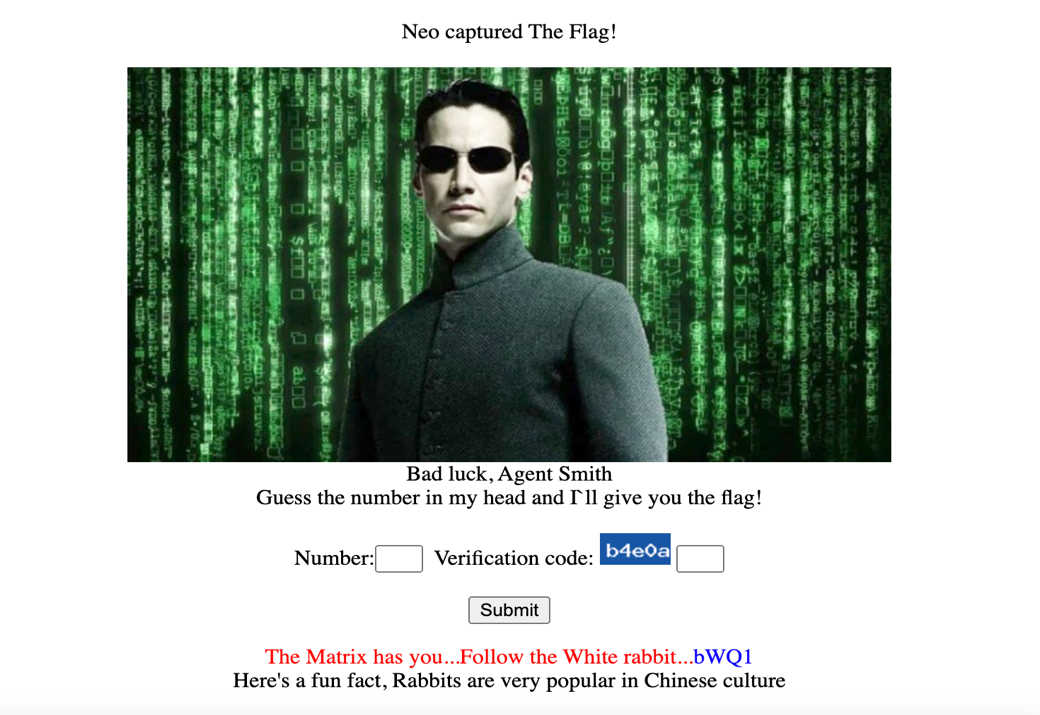
**ZXH CTF Report**

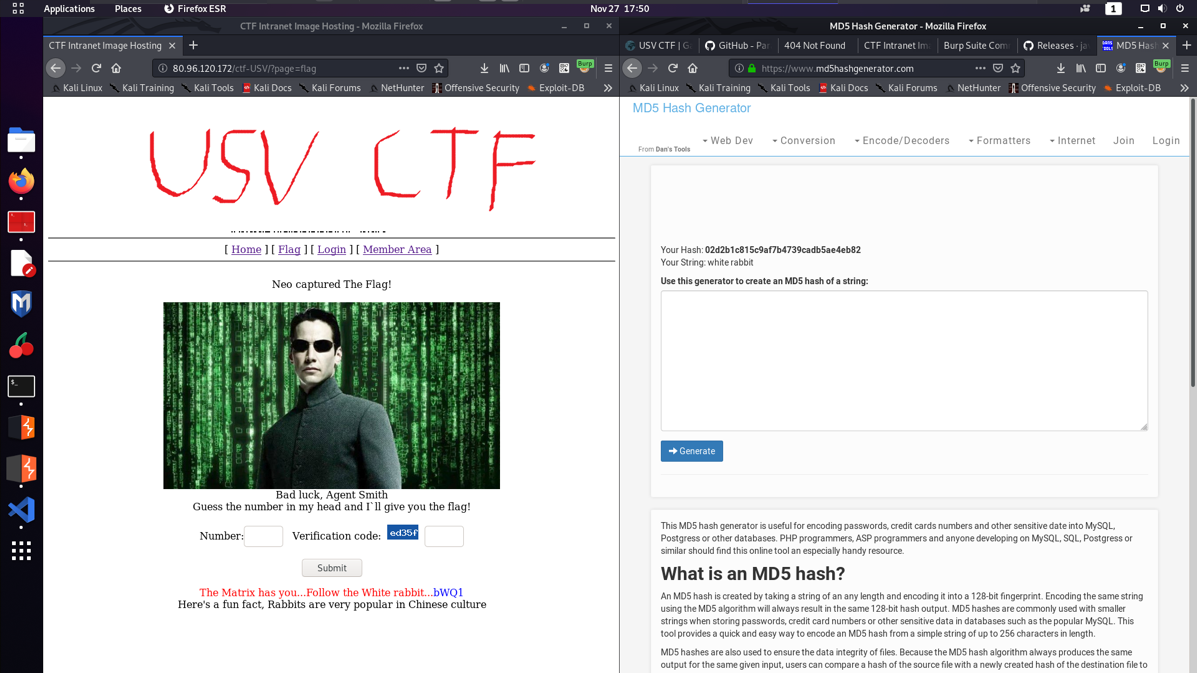
1. China challenge (30 points)

When we look over the <http://80.96.120.172/ctf-USV/?page=flag> we get two fields, one with number input and one with captcha validation. At first we tried to make a dummy request, setting the number to one and the captcha to what it was. We got back this “Oh, I forgot to mention, Agent htimS, Smith’s friend, broke the captcha” which contains both Smith and hitmS (Smith spelled backwards ).

This was a clear hint towards writing the captcha backwards. Now with the number set to 1 and captcha written down backwards we get the following screen

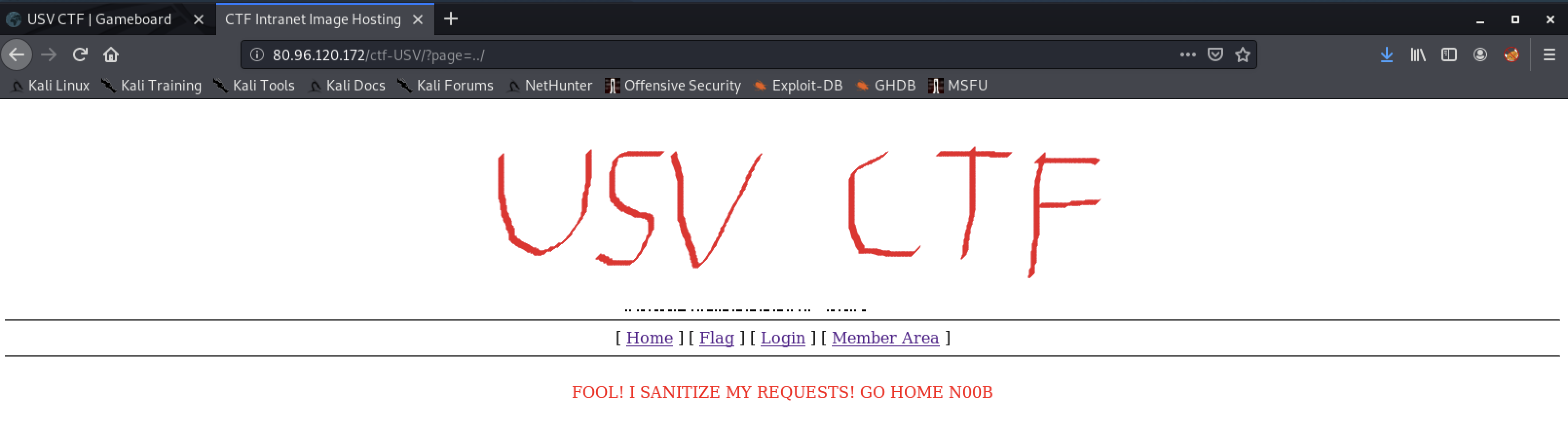


Here we have “bWQ1” which is a base64 encoded string. From the decode we get “md5”. So if we have “White rabbit” and md5 we tried to encrypt the “White rabbit” into md5 and put it as flag. But there was no luck, so we tried to make them “WHITE RABBIT” and “white rabbit” and we got a hit with the last one. The flag is: **02d2b1c815c9af7b4739cadb5ae4eb82**

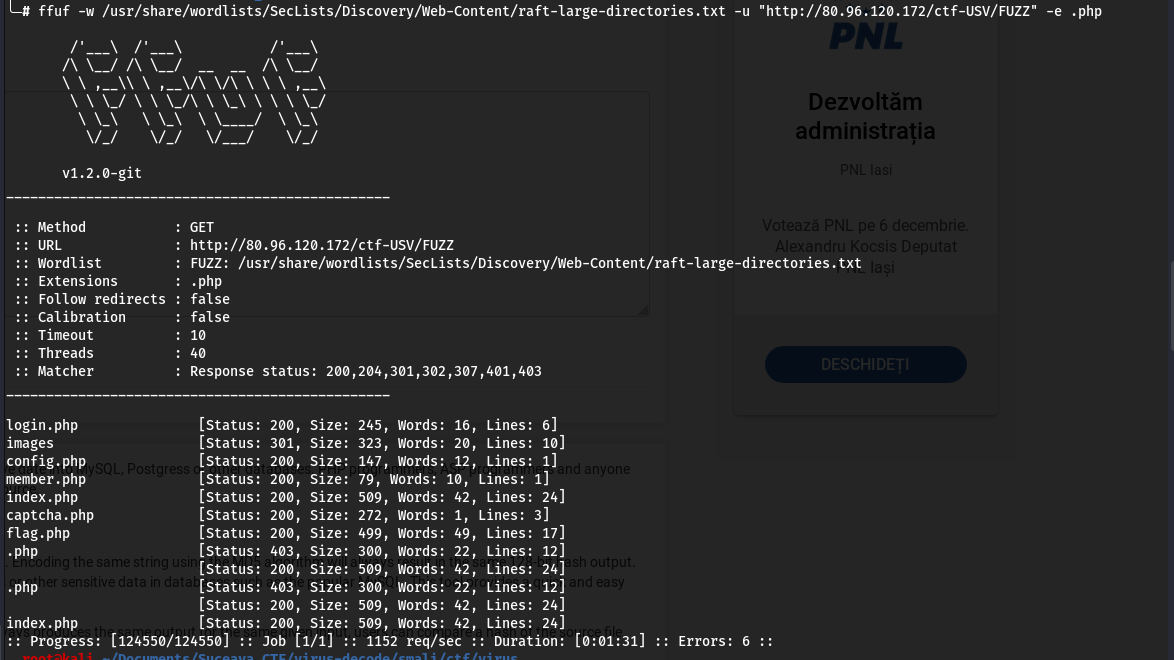


1. Greece challenge (60 points)

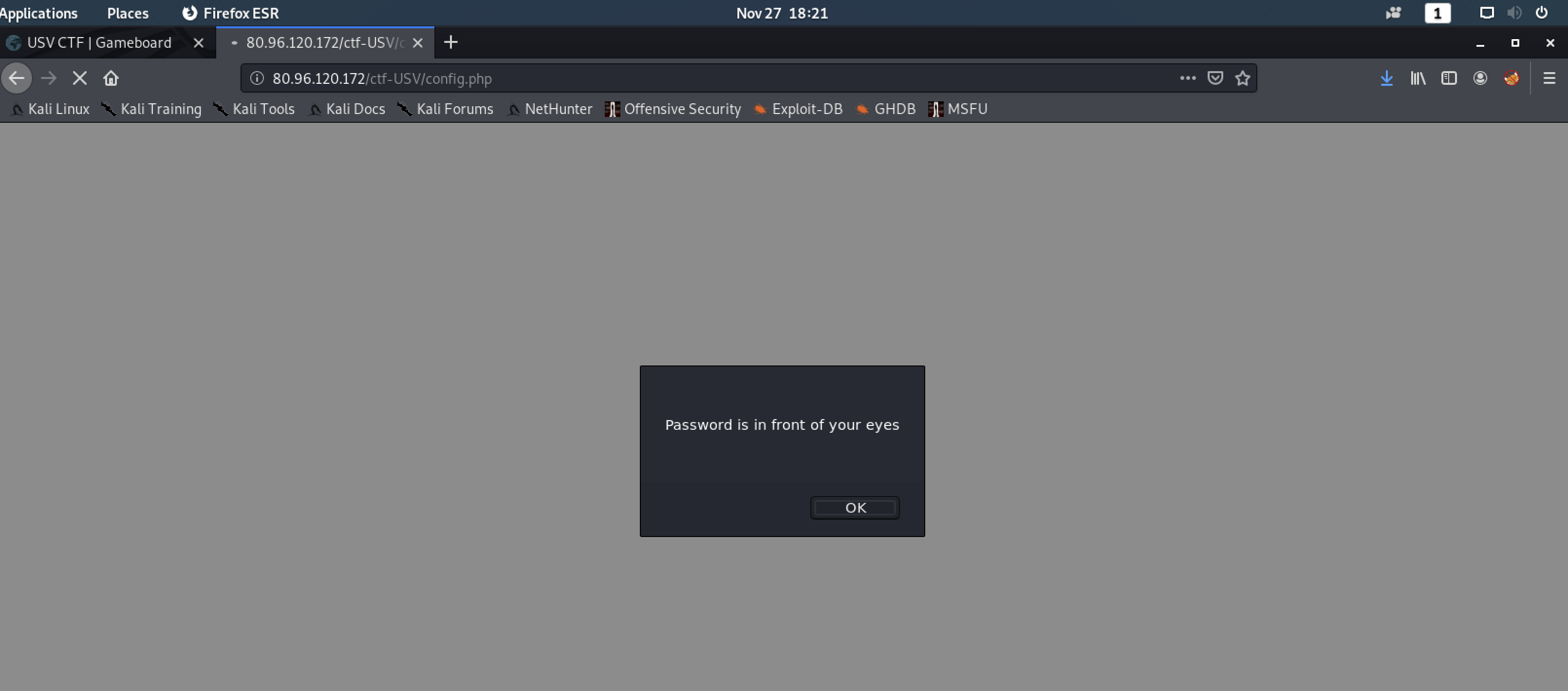
If we start to mess around with the page parameter we get that all special characters (such as “:;\/.”) are banned and instead of a valid request, we get the following message:



This started to made us think about a way to bypass the local file inclusion. But first we wanted to see what other files are on the machine, so we’ve started to enum using ffuf. The command that we’ve used is `ffuf -w /usr/share/wordlists/SecLists/Discovery/Web-Content/raft-large-directories.txt -u "http://80.96.120.172/ctf-USV/FUZZ" -e .php` which generated the following output:



From this we get a new file, config.php which is not linked anywhere on the website. If we try to access we get this output



If we click ok we get redirected to a blank page. Looking at the sourcecode there is revealed the flag: **cGdzOk5GMzRhc3RMNyhzdHV0dXNY**

