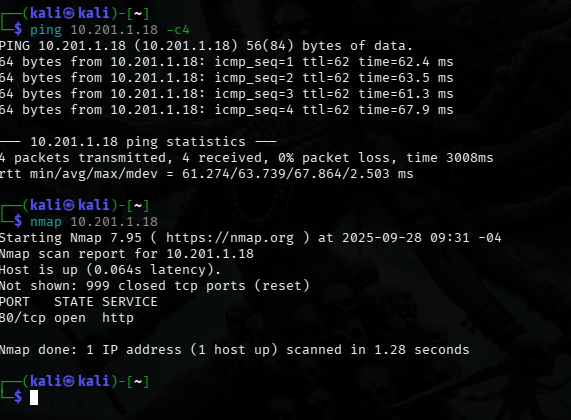
As the machine’s description indicates, we will be facing a potential IDOR vulnerability.

In case you don’t know, Insecure Direct Object References (IDORs) represent a form of access control flaw that occurs when a software application employs client-provided data to directly retrieve resources or data objects. While IDOR vulnerabilities are most frequently connected with horizontal privilege escalation (accessing another user's data at the same privilege level), they can also manifest in situations leading to vertical privilege escalation (gaining access to higher-level administrative functions). They can be also related to static objects (a single file or even a text string) as well as with variable data such as database objects.

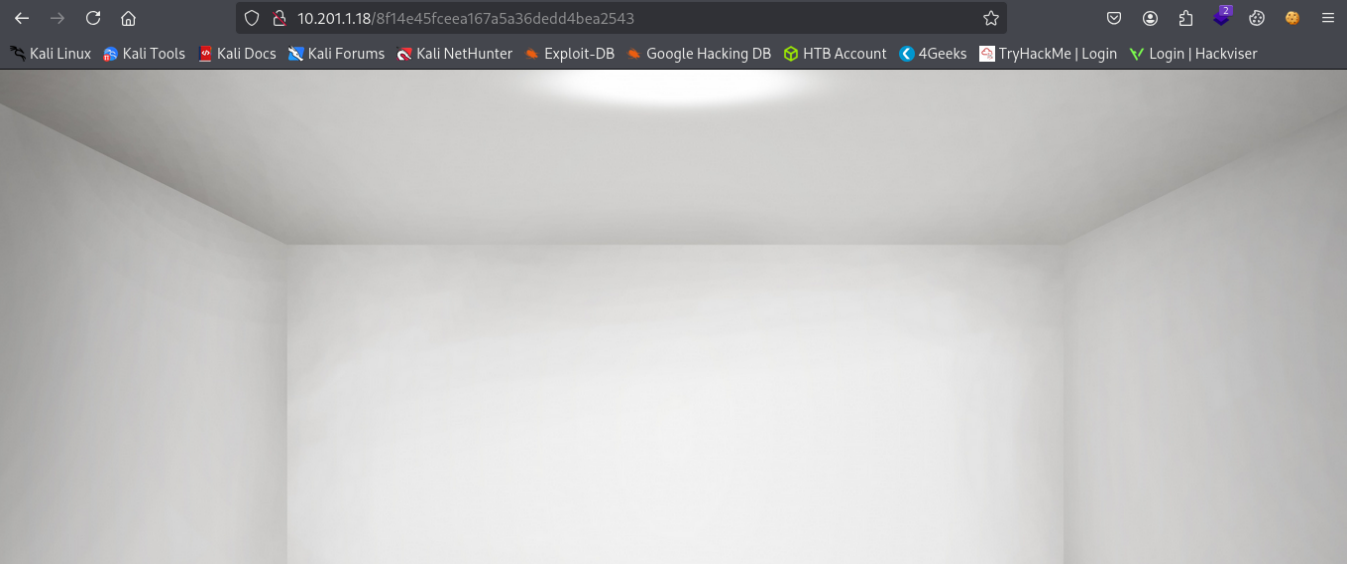
Let’s check this case. For instructional purposes I’ll keep the assigned IP and cover the results that lead directly to the solution. Let’s activate the machine, make sure we’re connected and make a first scan to analyze our target.



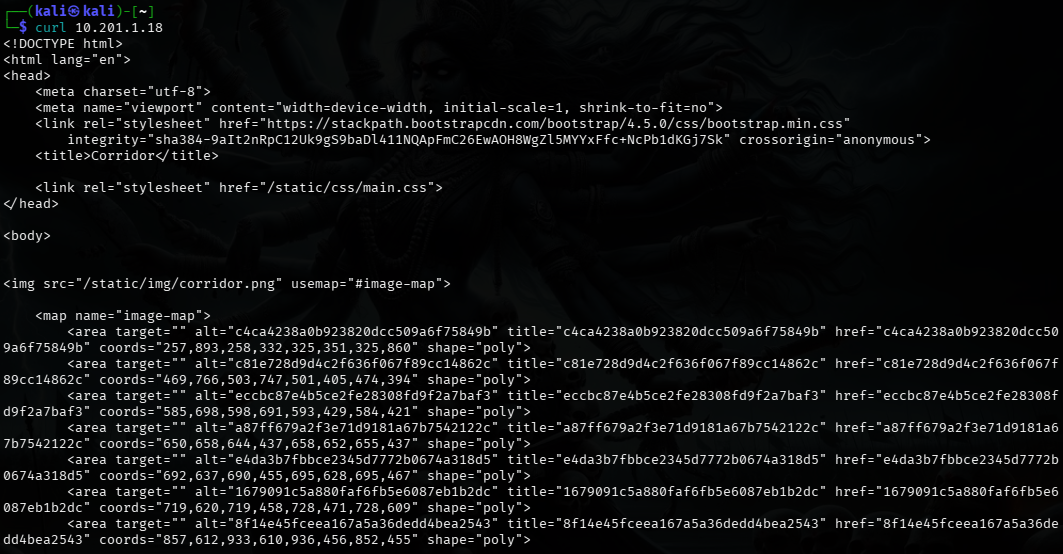
The results of our nmap primary scan tells us the only active service is http (p. 80) so we check it on our navigator to see what’s about.



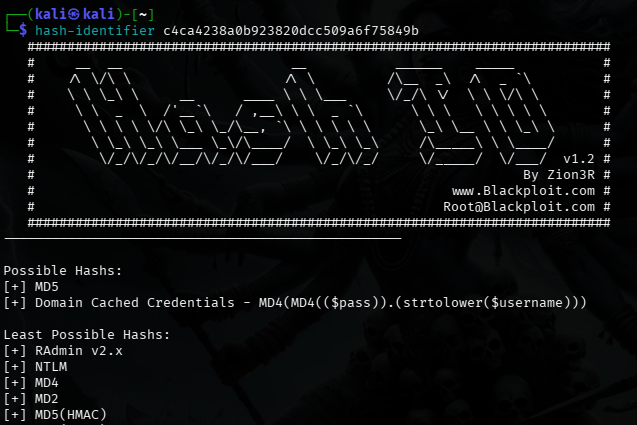
Interesting, there’s no plain text on the page, just the image of an corridor with 13 doors and a link in every one of them leading to…



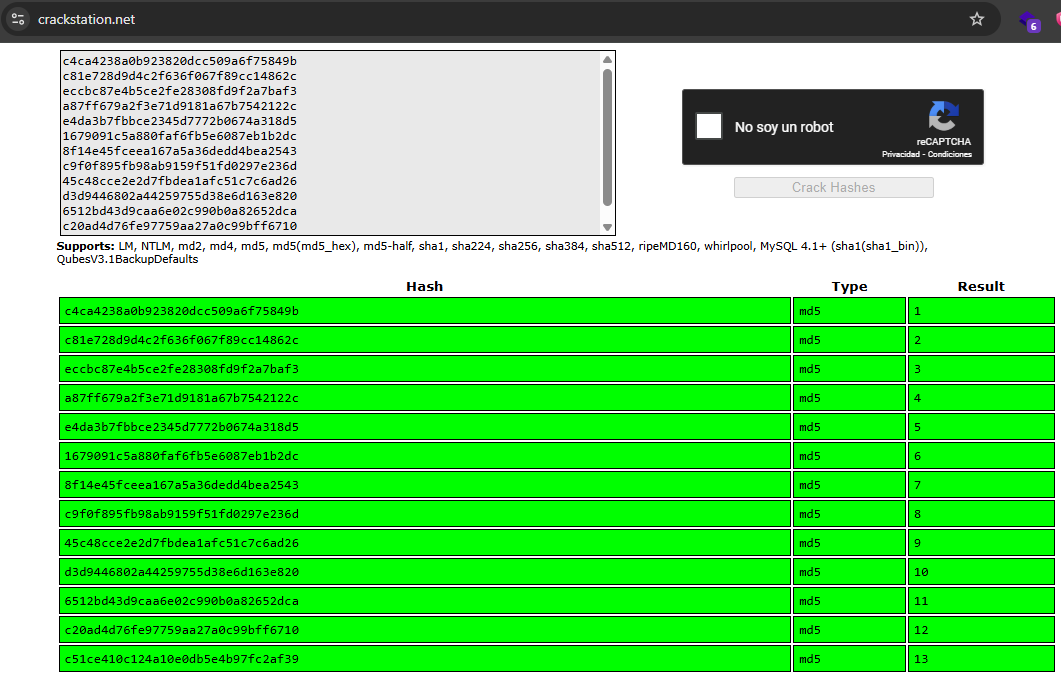
A white empty room, not the exit we are looking for. What are we missing here? Let’s search further. We can check on the page’s source code, but I choose to do so by typing curl (target\_ip) on a terminal and see it.



The “links” that look more like hashes are displayed in an specific order, what are we going to do now is to check what kind of hashes are them and why they’re displayed in that order, perhaps that’s the key. For that purpose I’m using the hash-identifier tool.

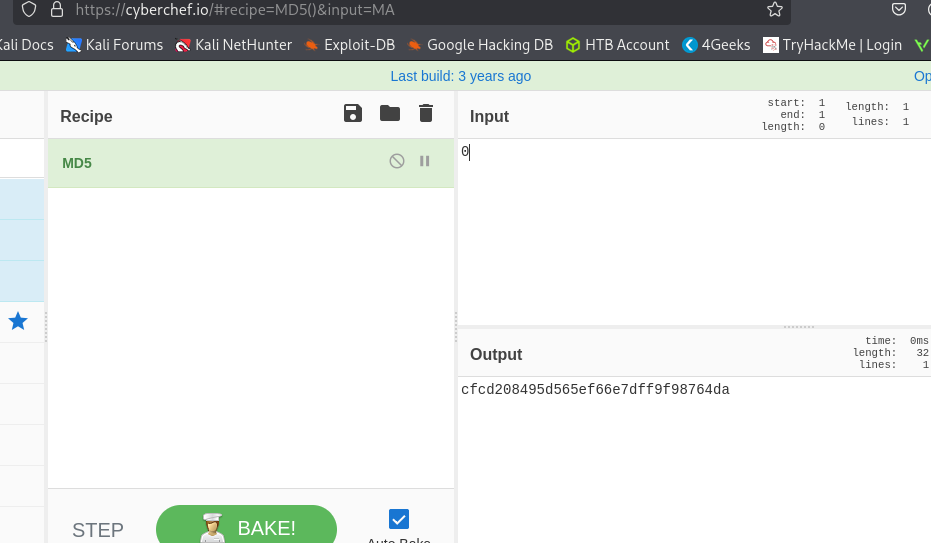


As I did it with the first one, I did with the rest of them. The tool says they’re most likely to be hashed under MD5 and that’s right because of the length, composed by 32 hexadecimal digits (0-9 and a-f). So I grab all of them and make a list to decrypt them. I did it with crackstation.net

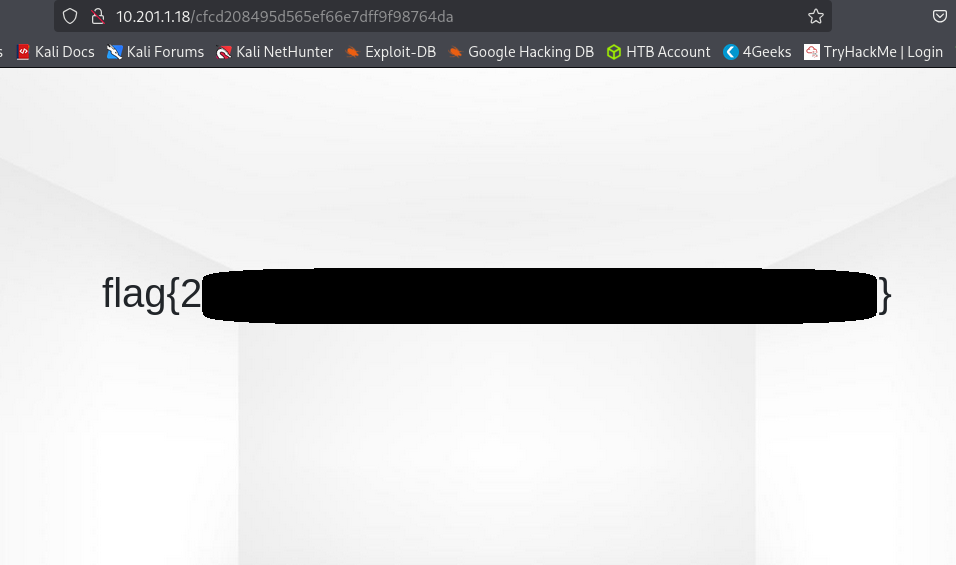


Interesting. It displays a sequence from 1 to 13, and as we saw before, if we click on each of these doors we see a ‘blank’ room, the hashes were the pages on every door, the exit we’re looking for might have the same format. But so far we don’t know which one is. But could it be another number we haven’t see?

When we walk into a corridor with 13 doors and we’re looking for the way back, do we go forward or backward? If the 1 is the number ahead, then we’re looking for the 0 if that’s the way back. Let’s go and figure it out. To keep the format, let’s find the MD5 hash for 0. I used cyberchef.io for it.



With this hash, we’re adding it up to the target ip in our navigator to find out and then…



…the flag appears before us. And that’s it. Not with complex commands nor sophisticated techniques, just common sense and a bit of basic knowledge of simple commands and regular tools.