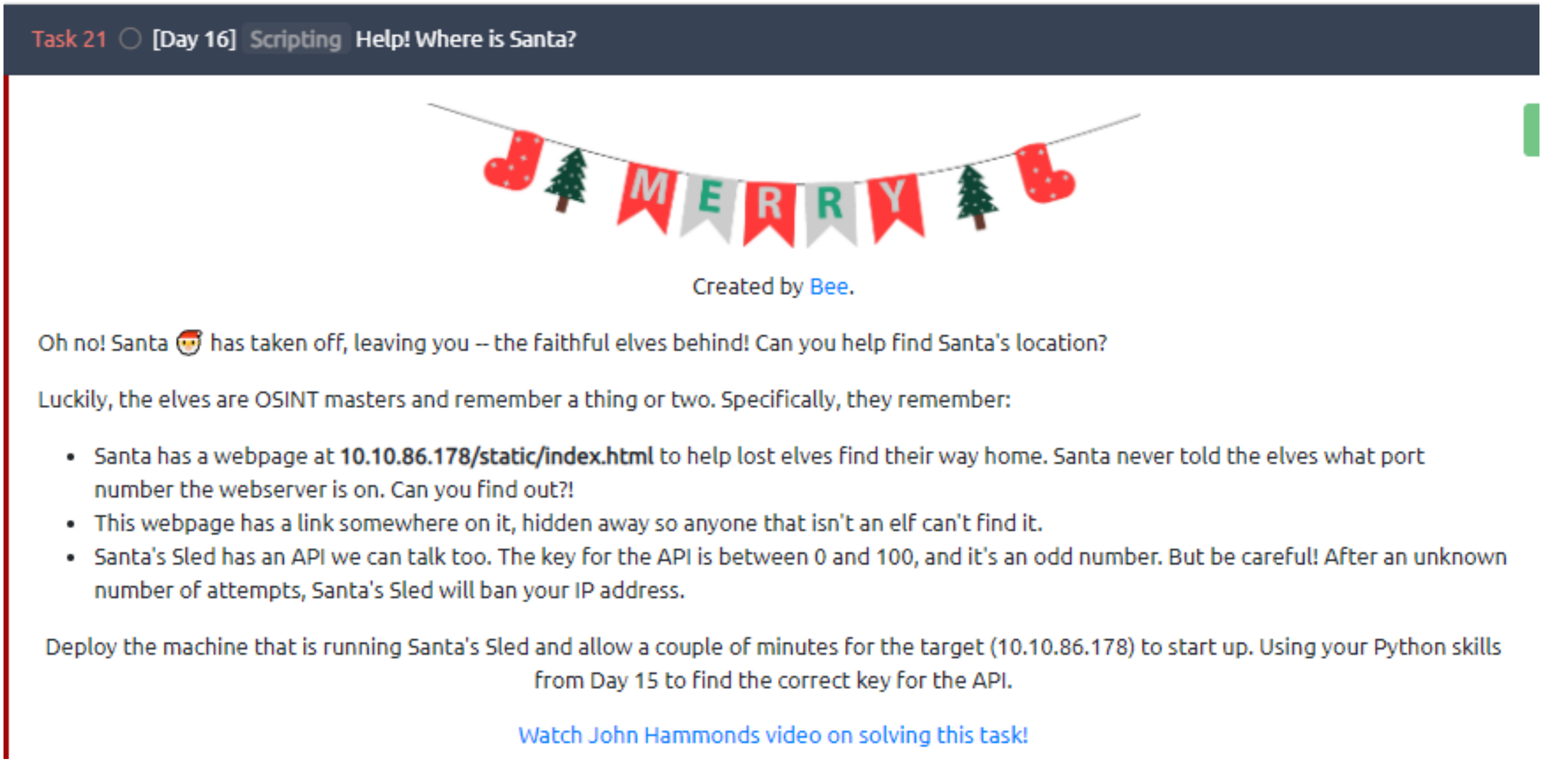
Day 16 - Help! Where is Santa?

Scenario



Task 21 ○ [Day 16] Scripting Help! Where is Santa?

Created by Bee.

Oh no! Santa 🎅 has taken off, leaving you -- the faithful elves behind! Can you help find Santa's location?

Luckily, the elves are OSINT masters and remember a thing or two. Specifically, they remember:

- Santa has a webpage at **10.10.86.178/static/index.html** to help lost elves find their way home. Santa never told the elves what port number the webserver is on. Can you find out?!
- This webpage has a link somewhere on it, hidden away so anyone that isn't an elf can't find it.
- Santa's Sled has an API we can talk too. The key for the API is between 0 and 100, and it's an odd number. But be careful! After an unknown number of attempts, Santa's Sled will ban your IP address.

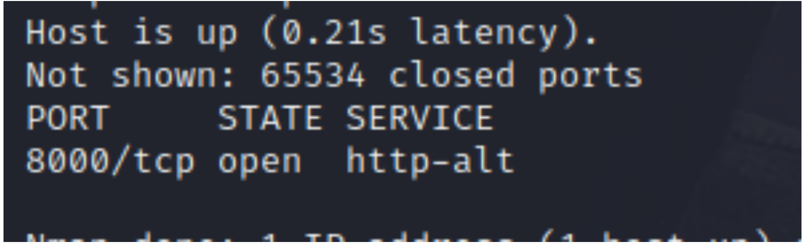
Deploy the machine that is running Santa's Sled and allow a couple of minutes for the target (10.10.86.178) to start up. Using your Python skills from Day 15 to find the correct key for the API.

Watch John Hammonds video on solving this task!

this challenge are tend to train our python scripting skills

let's scan for the web server port first, we can use nmap to do it

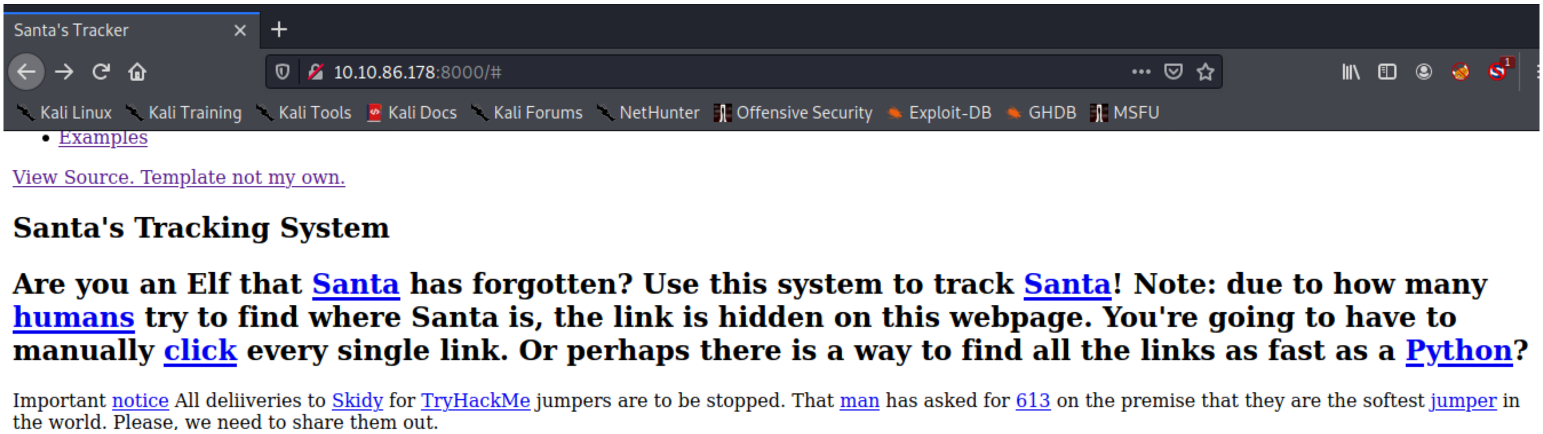
we've found the web server port on 8000



```
Host is up (0.21s latency).
Not shown: 65534 closed ports
PORT     STATE SERVICE
8000/tcp open  http-alt
```

Question: What is the port number for the web server?
-8000

checking out the root page of the web server



- Examples

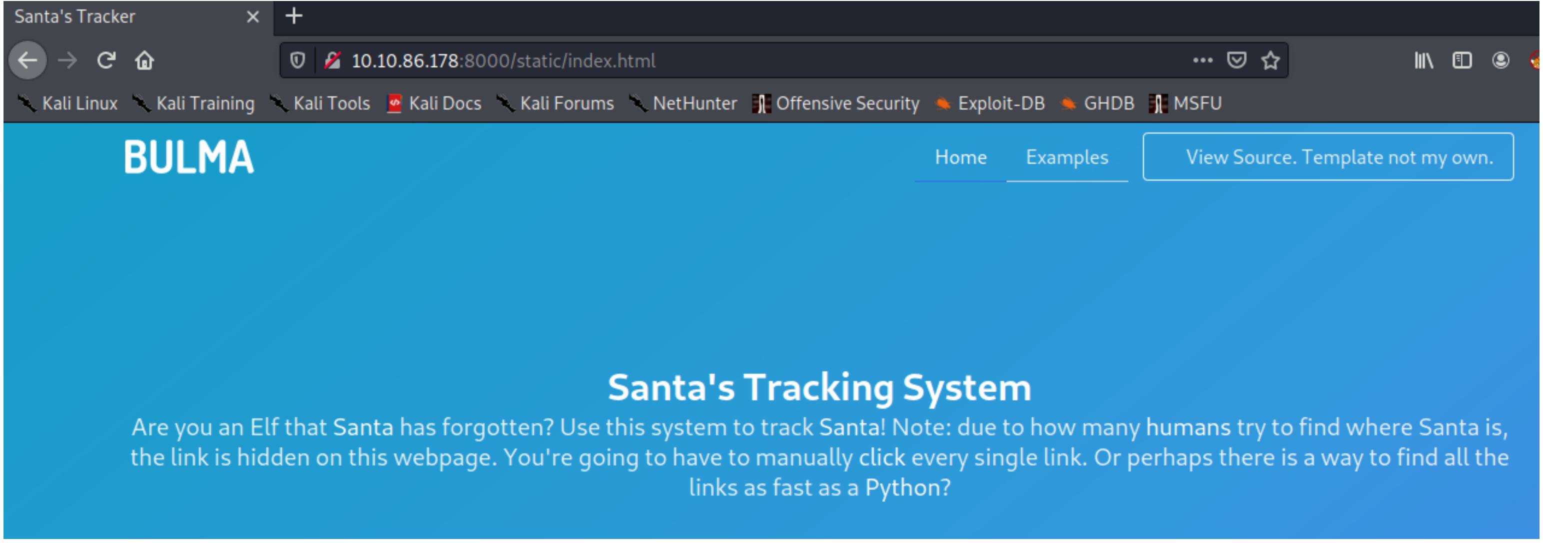
View Source. Template not my own.

Santa's Tracking System

Are you an Elf that Santa has forgotten? Use this system to track Santa! Note: due to how many humans try to find where Santa is, the link is hidden on this webpage. You're going to have to manually click every single link. Or perhaps there is a way to find all the links as fast as a Python?

Important notice All deliiveries to Skidy for TryHackMe jumpers are to be stopped. That man has asked for 613 on the premise that they are the softest jumper in the world. Please, we need to share them out.

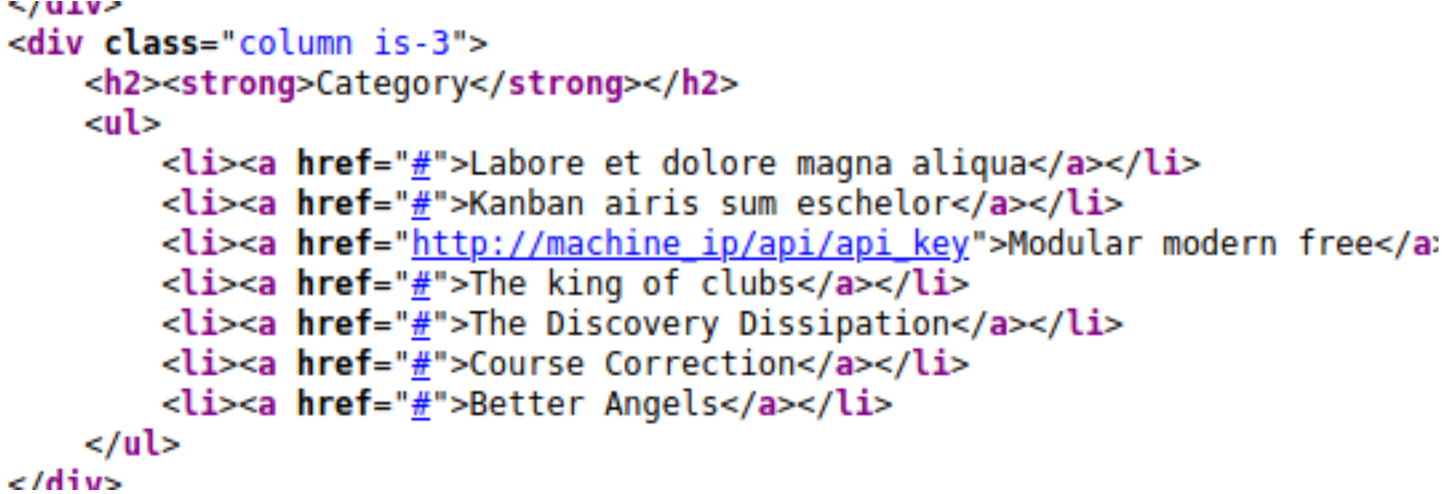
now let's check out the link that shown in the challenge desc, it looks nicer now



Santa's Tracking System

Are you an Elf that Santa has forgotten? Use this system to track Santa! Note: due to how many humans try to find where Santa is, the link is hidden on this webpage. You're going to have to manually click every single link. Or perhaps there is a way to find all the links as fast as a Python?

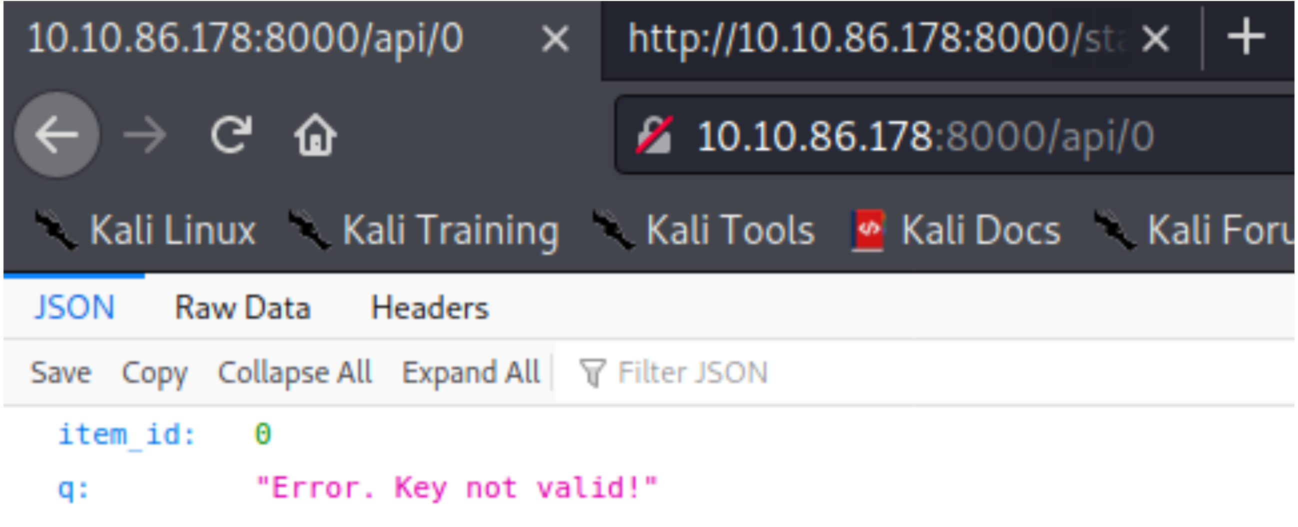
let's enumerate the web source code to see is there anything interesting for us?
// /api/api_key directory? that looks interesting here



```
</div>
<div class="column is-3">
    <h2><strong>Category</strong></h2>
    <ul>
        <li><a href="#">Labore et dolore magna aliqua</a></li>
        <li><a href="#">Kanban airis sum eschelor</a></li>
        <li><a href="http://machine_ip/api/api_key">Modular modern free</a
        <li><a href="#">The king of clubs</a></li>
        <li><a href="#">The Discovery Dissipation</a></li>
        <li><a href="#">Course Correction</a></li>
        <li><a href="#">Better Angels</a></li>
    </ul>
</div>
```

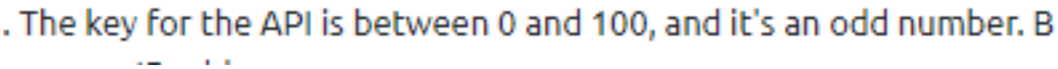
Question: Without using enumerations tools such as Dirbuster, what is the directory for the API? (without the API key)
- /api/

so now we've found the api directory but we need to find the api_key value, the api value will be 0-100, let's write our own python script to do it notes: if it's a wrong api key it should return “Error, Key not valid!”



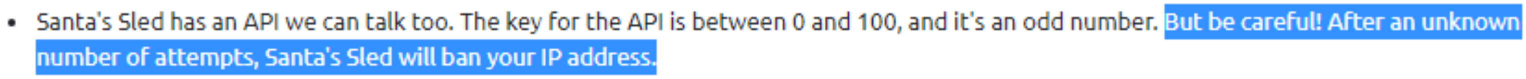
```
item_id:    0
q:          "Error. Key not valid!"
```

the API should be an odd number



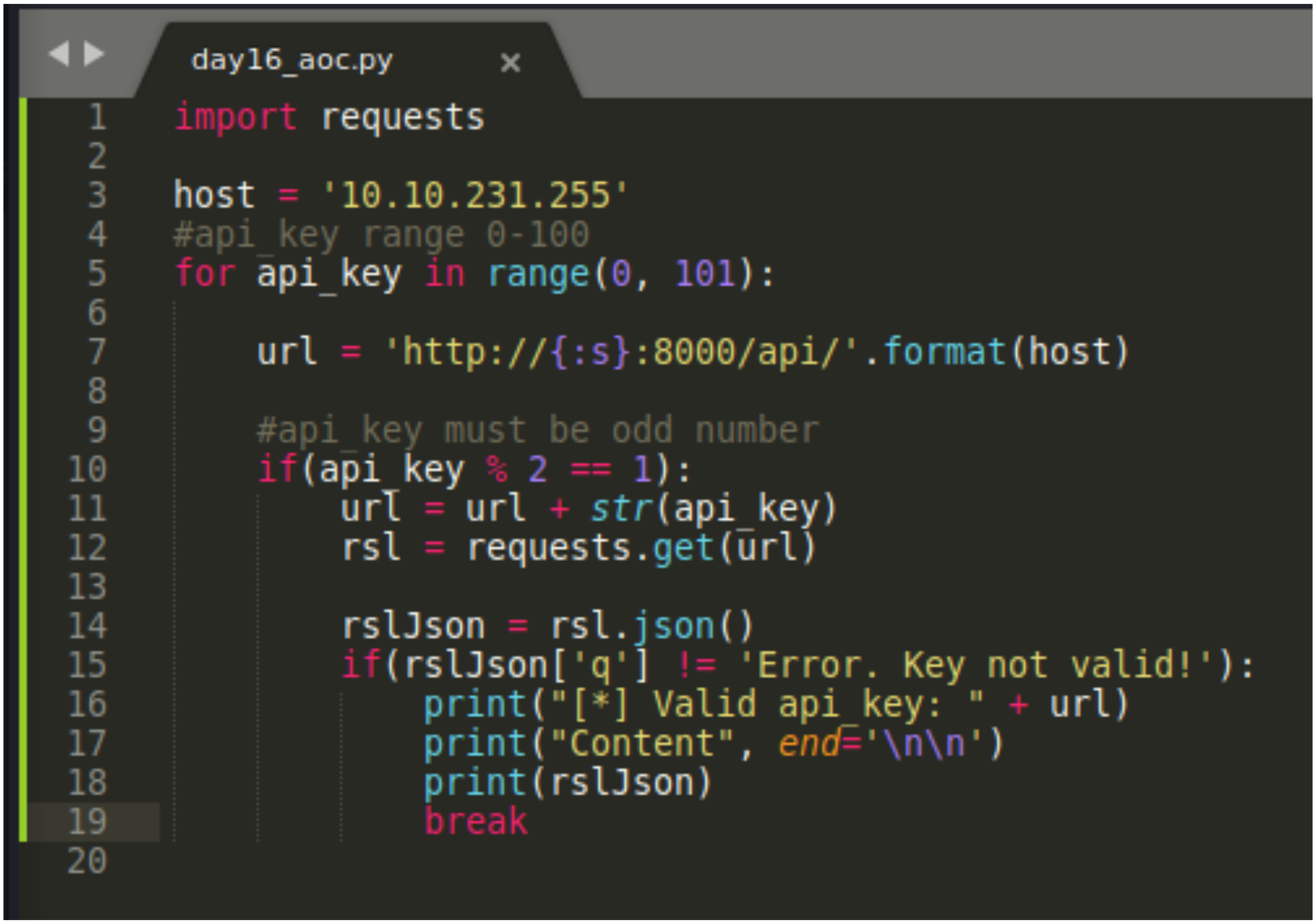
. The key for the API is between 0 and 100, and it's an odd number. B

but the problem is that, our IP will be banned for several unknown attempt



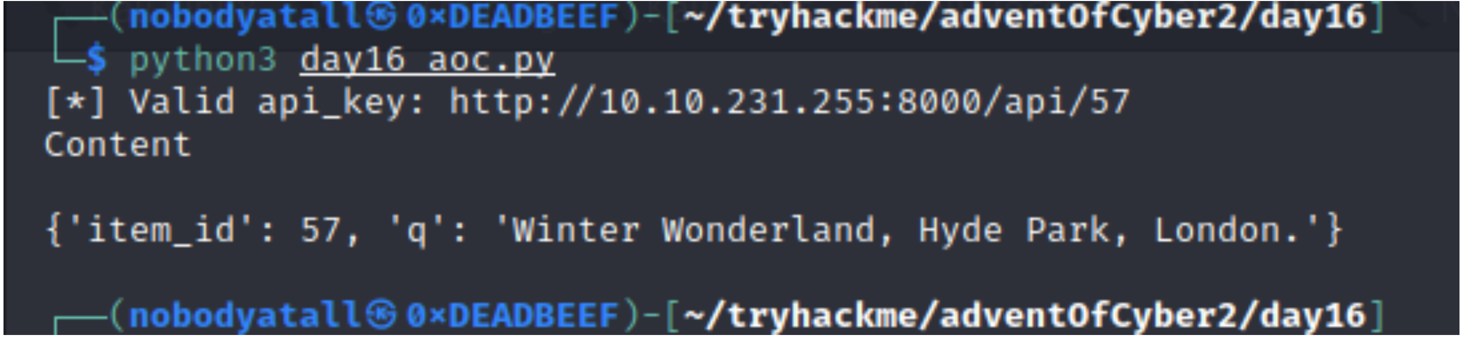
- Santa's Sled has an API we can talk too. The key for the API is between 0 and 100, and it's an odd number. But be careful! After an unknown number of attempts, Santa's Sled will ban your IP address.

now we've all the intel, let's write our python script by ignoring that banning IP issue first see whether it works or not



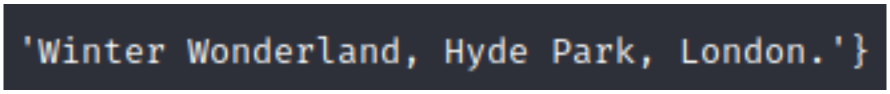
```python
import requests

host = '10.10.231.255'
#api_key range 0-100
for api_key in range(0, 101):

    url = 'http://{:s}:8000/api/'.format(host)

    #api_key must be odd number
    if(api_key % 2 == 1):
        url = url + str(api_key)
        rsl = requests.get(url)

        rslJson = rsl.json()
        if(rslJson['q'] != 'Error. Key not valid!'):
            print("[*] Valid api key: " + url)
            print("Content", end='\n\n')
            print(rslJson)
            break

```

executing it & we found the valid API ‘57’



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/adventOfCyber2/day16]
└─$ python3 day16_aoc.py
[*] Valid api_key: http://10.10.231.255:8000/api/57
Content

{'item_id': 57, 'q': 'Winter Wonderland, Hyde Park, London.'}

┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/adventOfCyber2/day16]
```

santa is at this location



```
'Winter Wonderland, Hyde Park, London.'}
```

so my conclusion, most probably the banning IP mechanism will be activate if we did not find the valid API_KEY with odd numbers since if we used 0-100, it'll be 101 attempts

so if we only enumerate the odd numbers it would be 50 attempts only