

PSP0201

Week 5

Writeup

Group Name: Undecided

Members

ID	Name	Role
1211101390	Aslamia Najwa Binti Ahmad Khadri	Leader
1211100431	Mohammad Omar Torofder	Member
1211103388	Vishnu Karmegam	Member
1211103092	Farryn Aisha binti Muhd Firdaus	Member

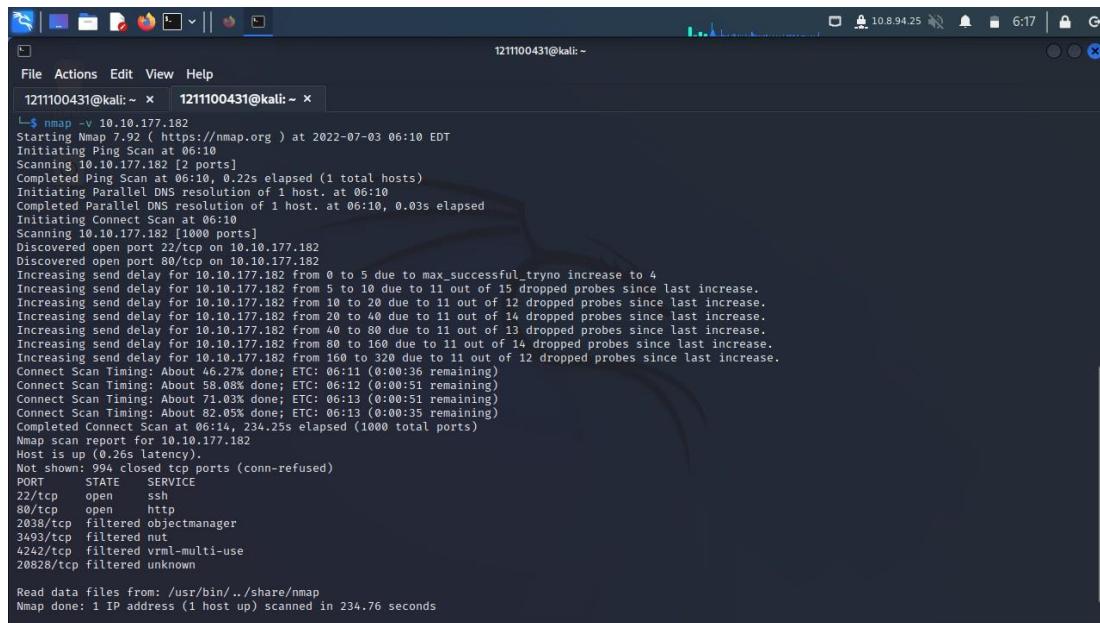
Day 16 : Scripting – Help! Where is Santa?

Tools used: Attackbox, Firefox

Solution/walkthrough:

Question 1

By using the nmap command in the terminal we found the open port which is 22 and 80 as shown in the picture from the list of port, state, and service.

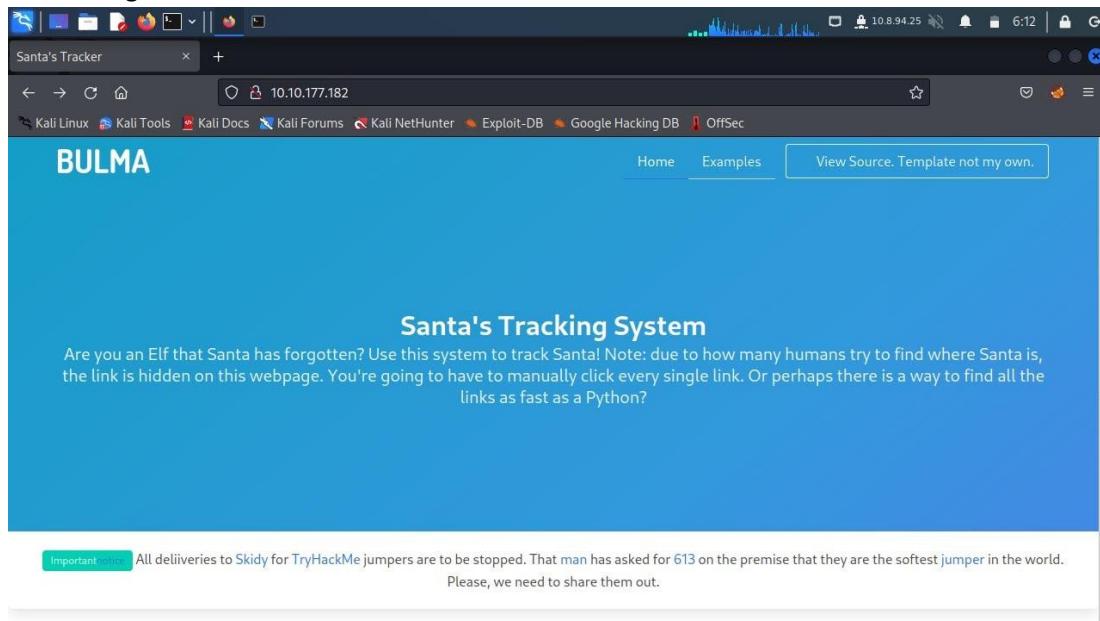


```
1211100431@kali: ~ x 1211100431@kali: ~
└$ nmap -v 10.10.177.182
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-03 06:10 EDT
Initiating Ping Scan at 06:10
Scanning 10.10.177.182 [2 ports]
Completed Ping Scan at 06:10, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host at 06:10
Completed Parallel DNS resolution of 1 host at 06:10, 0.03s elapsed
Initiating Connect Scan at 06:10
Completed Connect Scan at 06:10
Scanning 10.10.177.182 [1000 ports]
Discovered open port 22/tcp on 10.10.177.182
Discovered open port 80/tcp on 10.10.177.182
Increasing send delay for 10.10.177.182 from 0 to 5 due to max_successful_tryno increase to 4
Increasing send delay for 10.10.177.182 from 5 to 10 due to 11 out of 15 dropped probes since last increase.
Increasing send delay for 10.10.177.182 from 10 to 20 due to 11 out of 12 dropped probes since last increase.
Increasing send delay for 10.10.177.182 from 20 to 40 due to 11 out of 14 dropped probes since last increase.
Increasing send delay for 10.10.177.182 from 40 to 80 due to 11 out of 13 dropped probes since last increase.
Increasing send delay for 10.10.177.182 from 80 to 160 due to 11 out of 14 dropped probes since last increase.
Increasing send delay for 10.10.177.182 from 160 to 320 due to 11 out of 12 dropped probes since last increase.
Connect Scan Timing: About 46.27% done; ETC: 06:11 (0:00:36 remaining)
Connect Scan Timing: About 58.08% done; ETC: 06:12 (0:00:51 remaining)
Connect Scan Timing: About 71.03% done; ETC: 06:13 (0:00:51 remaining)
Connect Scan Timing: About 82.05% done; ETC: 06:13 (0:00:35 remaining)
Completed Connect Scan at 06:14, 234.25s elapsed (1000 total ports)
Nmap scan report for 10.10.177.182
Host is up (0.026s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    open       http
2038/tcp  filtered  objectmanager
3493/tcp  filtered  nut
4242/tcp  filtered  vrmr-multi-use
20828/tcp filtered  unknown

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 234.76 seconds
```

Question 2

We browse the IP address from the machine and was led to this website. Here, we found the templates that is being used.



Question 3

Next, we were ordered to find the api directory. We repurpose the python code that we had used before in day 15 to find the api directory.

The screenshot shows a terminal window with a dark theme. At the top, it displays the path '/usr/share/kali/tools/Google/LinkGrabber/linkgraber.py'. Below that is the command 'python3' followed by the script name. The terminal window title is 'linkgraber.py'. The script content is a Python program using BeautifulSoup to extract links from a webpage. It includes imports for requests and BeautifulSoup, sets the URL to 'http://10.10.10.181', parses the HTML with 'lxml', finds all href attributes, and prints each link. The terminal shows the script running and outputting several URLs.

```
GNU nano 6.2
#!/usr/bin/python3
# import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail

# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail
from bs4 import BeautifulSoup
import requests

# replace testurl.com with the url you want to use.
# requests.get downloads the webpage and stores it as a variable
html = requests.get('http://10.10.10.181')

# this parses the webpage into something that beautifulsoup can read over
soup = BeautifulSoup(html.text, "lxml")
# lxml is just the parser for reading the html

# this is the line that grabs all the links # stores all the links in the links variable
links = soup.find_all('a href')
for link in links:
    # prints each link
    print(link)
```

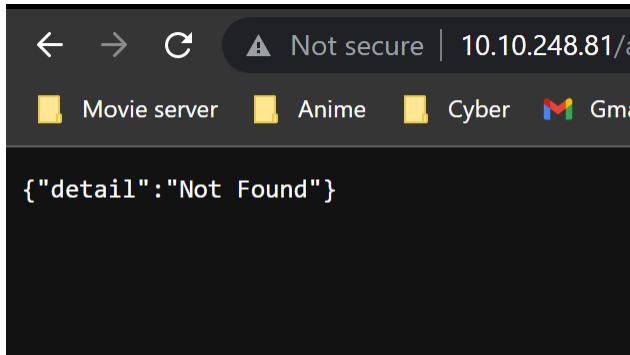
After running this python code, we receive the following output where we found the api directory.

The screenshot shows the terminal output of the script. It lists several URLs, including one that points to the 'api' directory at 'http://machine_ip/api/api_key'. This indicates that the 'api' directory exists on the machine.

```
<a href="#">Aisia caisia</a>
<a href="#">Murphy's law</a>
<a href="#">Flimsy Lavenrock</a>
<a href="#">Maven Mousie Lavender</a>
<a href="#">Labore et dolore magna aliqua</a>
<a href="#">Kanban airis sum eschelor</a>
<a href="http://machine_ip/api/api_key">Modular modern free</a>
<a href="#">The king of clubs</a>
```

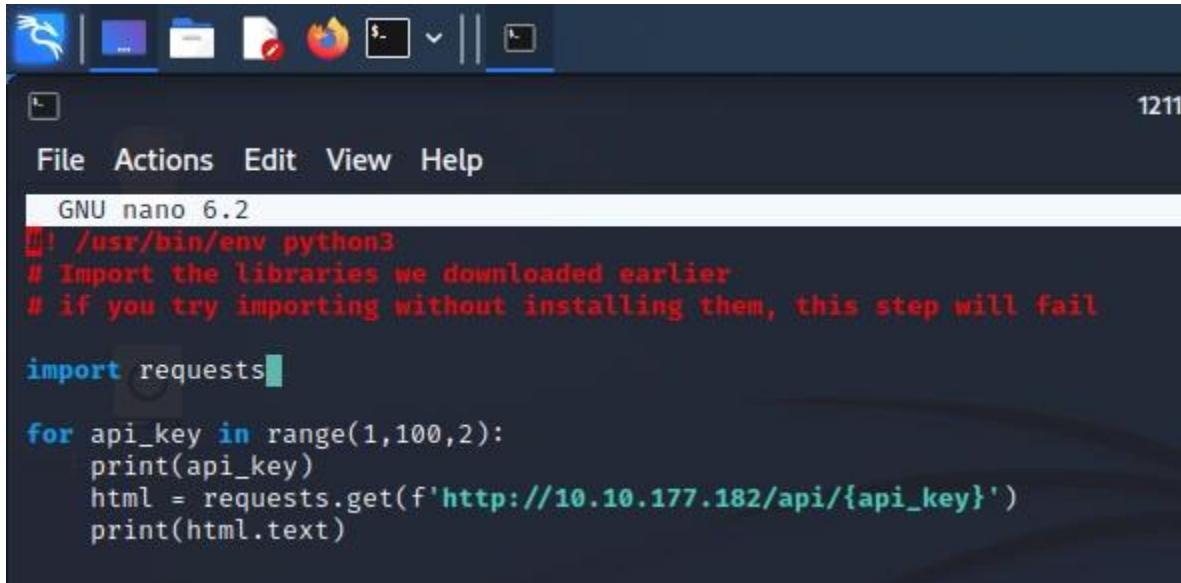
Question 4

We are asked to check what do we get if we don't add any parameters after api.



Question 5 & 6

Here we do know the api directory but not the api key, so now we need to find the api codes to find Santa's location. However, we are told that api key is between 0-100 and it's an odd number. So, we can use the codes again to automate the process.



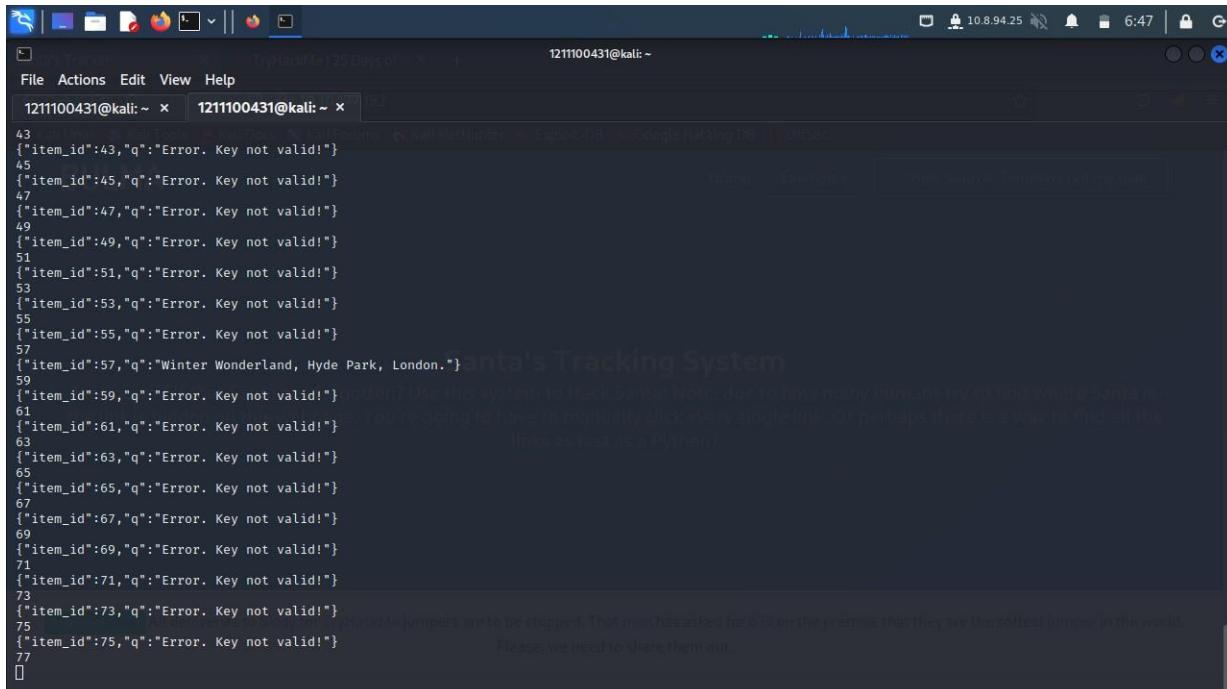
The screenshot shows a terminal window with a dark blue header bar containing icons for file operations and a clock showing 12:11. Below the header is a menu bar with "File", "Actions", "Edit", "View", and "Help". The main area of the terminal is a text editor titled "GNU nano 6.2" displaying the following Python code:

```
#!/usr/bin/env python3
# Import the libraries we downloaded earlier
# if you try importing without installing them, this step will fail

import requests

for api_key in range(1,100,2):
    print(api_key)
    html = requests.get(f'http://10.10.177.182/api/{api_key}')
    print(html.text)
```

When we run this, we got the result as shown, you can see code 57 is the key and Santa is currently in Winter Wonderland, Hyde Park, London.



The screenshot shows a terminal window with a dark blue header bar containing icons for file operations and a clock showing 6:47. Below the header is a menu bar with "File", "Actions", "Edit", "View", and "Help". The main area of the terminal shows the output of the Python script, which lists odd-numbered API keys from 1 to 99. The output for key 57 is highlighted in yellow:

```
43 {"item_id":43,"q":"Error. Key not valid!"}
45 {"item_id":45,"q":"Error. Key not valid!"}
47 {"item_id":47,"q":"Error. Key not valid!"}
49 {"item_id":49,"q":"Error. Key not valid!"}
51 {"item_id":51,"q":"Error. Key not valid!"}
53 {"item_id":53,"q":"Error. Key not valid!"}
55 {"item_id":55,"q":"Error. Key not valid!"}
57 {"item_id":57,"q":"Winter Wonderland, Hyde Park, London."}
59 {"item_id":59,"q":"Error. Key not valid!"}
61 {"item_id":61,"q":"Error. Key not valid!"}
63 {"item_id":63,"q":"Error. Key not valid!"}
65 {"item_id":65,"q":"Error. Key not valid!"}
67 {"item_id":67,"q":"Error. Key not valid!"}
69 {"item_id":69,"q":"Error. Key not valid!"}
71 {"item_id":71,"q":"Error. Key not valid!"}
73 {"item_id":73,"q":"Error. Key not valid!"}
75 {"item_id":75,"q":"Error. Key not valid!"}
```

Thought Process/Methodology:

At the beginning we used nmap to find information about the IP, where we were able to find that ports 22 and 80 is open. When we added this in the URL with the IP address, we were redirected to a website called Santa's tracking system. On the website, we found the templates that is being used. Next, we were ordered to find the api directory. We repurpose the python code that we had used before in day 15 to find the api directory. Next, we checked for the output when we do not add any parameters after api in raw data. Afterward, to find Santa's position, we utilize the python code that we had earlier to find his location, and which is the api key to find Santa. Although there was a chance that the system would block us if there were too many wrong inputs, we were told that the api key is in between 0-100 and that it is an odd number so it narrows down our search significantly. After running the python code, we found that Santa was in Winter wonder, Hyde Park, London and the api number was 57.