Submitted by: IQRA JAWAD AHMED

Report: Onion Search Scraper (Python)

1. Objective

The purpose of this Python script is to **automate the scraping of search results from a dark web (.onion) search engine**.
The script takes multiple search keywords (e.g., *market*, *forum*, *bitcoin*), fetches corresponding results, extracts structured data, and saves it into a JSON file for further analysis.

2. Tools & Libraries Used

- **Python 3.13**
- **requests_html** → For sending HTTP requests and handling pages.
- **BeautifulSoup (bs4)** → For parsing and extracting specific HTML elements.
- **json** → For exporting the scraped results into a structured format (JSON).
- **Tor Proxy (SOCKS5)** → Requests are routed through Tor running on `127.0.0.1:9050`.

3. Workflow of the Script

Step 1: Setup Session & Proxy

```
session = HTMLSession()
proxies = {
    'http': 'socks5h://127.0.0.1:9050',
    'https': 'socks5h://127.0.0.1:9050'
}
```

A Tor proxy is defined to ensure anonymity and access to `.onion` domains.

Step 2: Define Search Terms

```
search_terms = ['market', 'forum', 'bitcoin']
```

A list of keywords is defined. The script will loop through each keyword.

Step 3: Fetch HTML for Each Search

```
url = f"http://<onion_address>/search/?q={term}"
r = session.get(url, proxies=proxies)
soup = BeautifulSoup(r.text, "html.parser")
```

For each keyword, the script requests the search page and loads the HTML into BeautifulSoup.

Step 4: Parse Search Results

Each search result is inside `<li class="result">`.
From each result, the following fields are extracted:

- **Title** → `<h4><a>` tag
- **URL** → `href` attribute
- **Description** → `<p>` tag
- **Cite (source)** → `<cite>` tag
- **Last Seen** → `<span class="lastSeen">`

Example parsing code:

```
title = li.h4.find("a").text.strip() if li.h4 and li.h4.find('a') else None
url_ = li.find('a')['href'] if li.find('a') else None
desc = li.find("p").text.strip() if li.find('p') else None
```

Step 5: Store Data in JSON Format

For each keyword, results are stored in a dictionary:

```
results[term].append({
    "title": title,
    "url": url_,
    "description": desc,
    "cite": cite,
    "last_seen": last_seen
})
```

At the end, the entire dictionary is dumped into `results.json`.

4. Sample Output (JSON)

```
{
    "market": [
        {
            "title": "Dark Market",
            "url": "http://xyz.onion/market",
            "description": "Buy and sell goods anonymously.",
            "cite": "xyz.onion",
            "last_seen": "2 days ago"
        }
    ],
    "forum": [
        {
            "title": "Hacker Forum",
            "url": "http://abc.onion/forum",
            "description": "Discussion about exploits and security.",
            "cite": "abc.onion",
            "last_seen": "1 day ago"
        }
    ]
}
```

5. Advantages of This Script

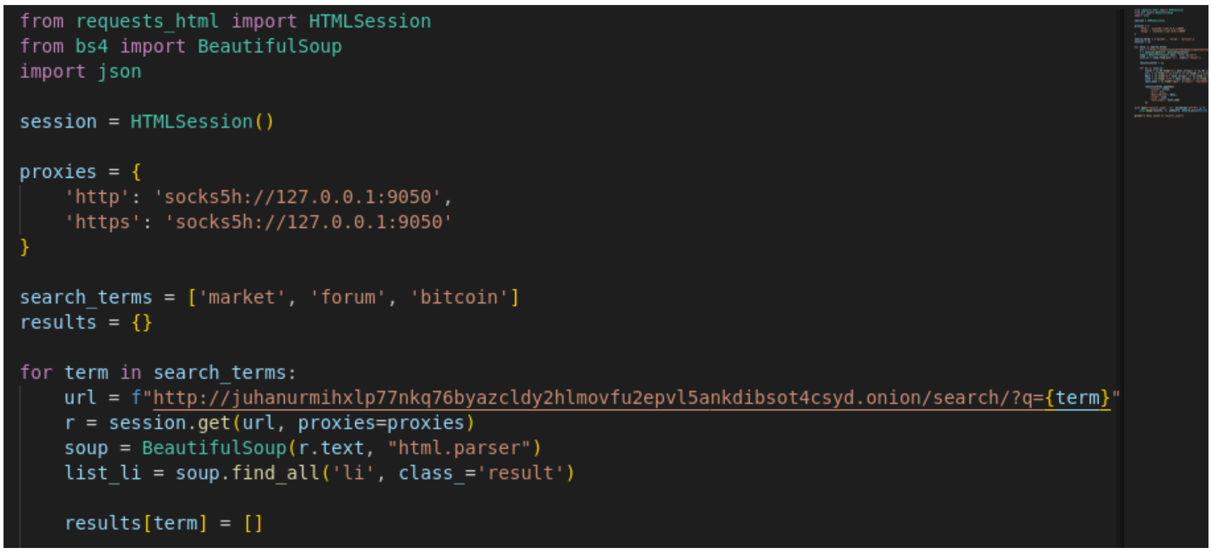
- **Automation** → Multiple keywords can be searched in one run.
- **Structured Data** → JSON format makes analysis easier.
- **Anonymity** → All requests go through Tor (SOCKS5).

- **Scalability** → Easy to add more keywords or save data to a database later.

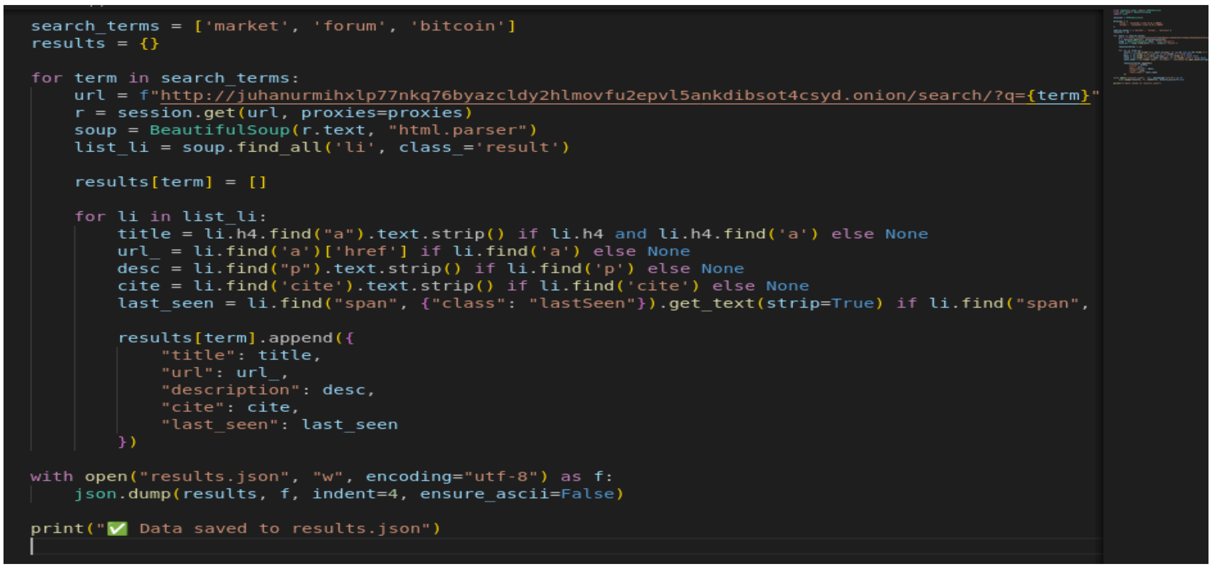
6. Limitations

- Requires Tor running on `127.0.0.1:9050`.
- If the target onion site uses heavy JavaScript, `requests_html.render()` may be needed.
- Scraping speed depends on Tor network latency.

CODE:

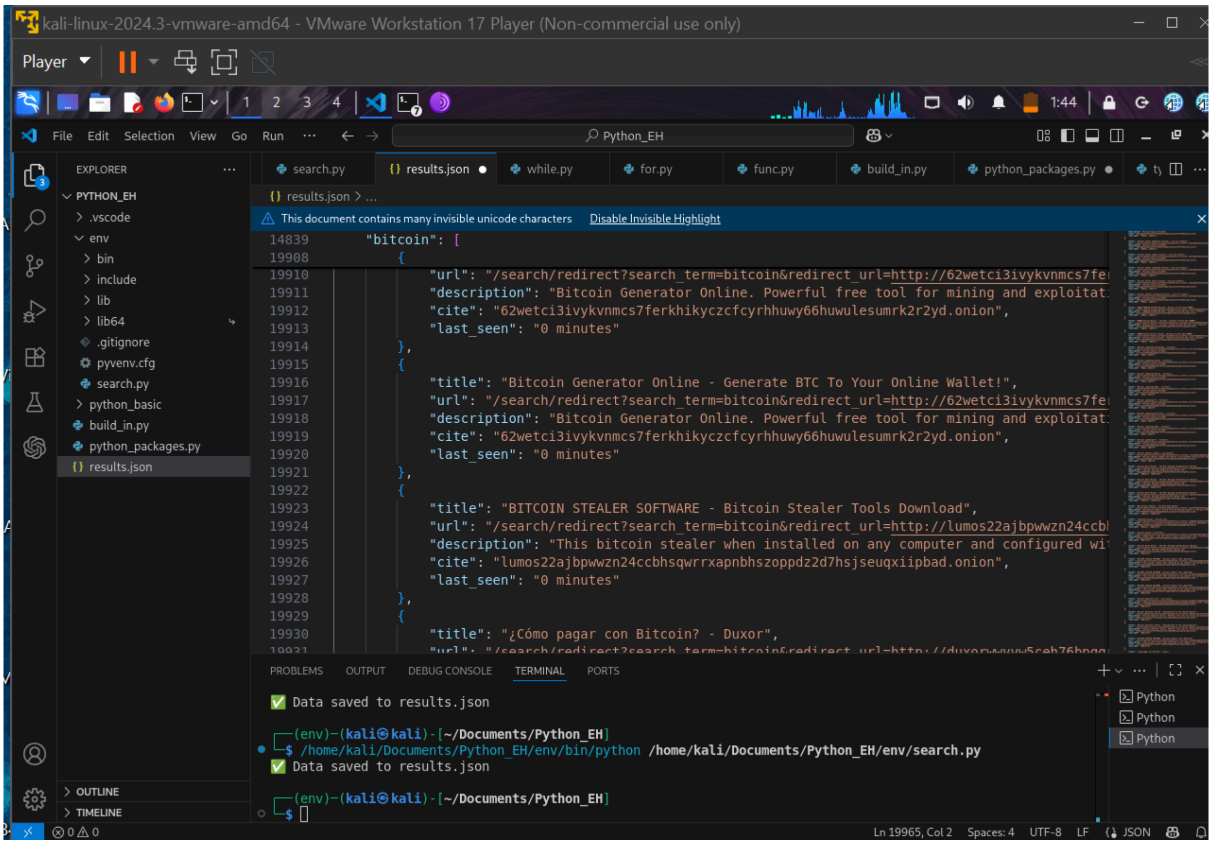


```python
from requests_html import HTMLSession
from bs4 import BeautifulSoup
import json

session = HTMLSession()

proxies = {
    'http': 'socks5h://127.0.0.1:9050',
    'https': 'socks5h://127.0.0.1:9050'
}

search_terms = ['market', 'forum', 'bitcoin']
results = {}

for term in search_terms:
    url = f"http://juhanurmihxlp77nkq76byazcldy2hlmovfu2epvl5ankdibsot4csyd.onion/search/?q={term}"
    r = session.get(url, proxies=proxies)
    soup = BeautifulSoup(r.text, "html.parser")
    list_li = soup.find_all('li', class_='result')

    results[term] = []
```



```python
search_terms = ['market', 'forum', 'bitcoin']
results = {}

for term in search_terms:
    url = f"http://juhanurmihxlp77nkq76byazcldy2hlmovfu2epvl5ankdibsot4csyd.onion/search/?q={term}"
    r = session.get(url, proxies=proxies)
    soup = BeautifulSoup(r.text, "html.parser")
    list_li = soup.find_all('li', class_='result')

    results[term] = []

    for li in list_li:
        title = li.h4.find("a").text.strip() if li.h4 and li.h4.find('a') else None
        url_ = li.find('a')['href'] if li.find('a') else None
        desc = li.find("p").text.strip() if li.find('p') else None
        cite = li.find('cite').text.strip() if li.find('cite') else None
        last_seen = li.find("span", {"class": "lastSeen"}).get_text(strip=True) if li.find("span",

        results[term].append({
            "title": title,
            "url": url_,
            "description": desc,
            "cite": cite,
            "last_seen": last_seen
        })

with open("results.json", "w", encoding="utf-8") as f:
    json.dump(results, f, indent=4, ensure_ascii=False)

print("✅ Data saved to results.json")
```

OUTPUT FILE:



```
    "bitcoin": [
        {
            "url": "/search/redirect?search_term=bitcoin&redirect_url=http://62wetci3ivykvnmcs7fe
            "description": "Bitcoin Generator Online. Powerful free tool for mining and exploitat
            "cite": "62wetci3ivykvnmcs7ferkhikyczcfcyrhhuwy66huwulesumrk2r2yd.onion",
            "last_seen": "0 minutes"
        },
        {
            "title": "Bitcoin Generator Online - Generate BTC To Your Online Wallet!",
            "url": "/search/redirect?search_term=bitcoin&redirect_url=http://62wetci3ivykvnmcs7fe
            "description": "Bitcoin Generator Online. Powerful free tool for mining and exploitat
            "cite": "62wetci3ivykvnmcs7ferkhikyczcfcyrhhuwy66huwulesumrk2r2yd.onion",
            "last_seen": "0 minutes"
        },
        {
            "title": "BITCOIN STEALER SOFTWARE - Bitcoin Stealer Tools Download",
            "url": "/search/redirect?search_term=bitcoin&redirect_url=http://lumos22ajbpwwzn24ccb
            "description": "This bitcoin stealer when installed on any computer and configured wi
            "cite": "lumos22ajbpwwzn24ccbhsqwrrxapnbhszoppdz2d7hsjseuqxiipbad.onion",
            "last_seen": "0 minutes"
        },
        {
            "title": "¿Cómo pagar con Bitcoin? - Duxor",
```

```
✅ Data saved to results.json

┌──(env)─(kali㉿kali)-[~/Documents/Python_EH]
└─$ /home/kali/Documents/Python_EH/env/bin/python /home/kali/Documents/Python_EH/env/search.py
✅ Data saved to results.json

┌──(env)─(kali㉿kali)-[~/Documents/Python_EH]
└─$ 
```