### Code 1: dark\_web\_monitor.py

import requests

from bs4 import BeautifulSoup

import re

import time

import urllib3

# Suppress only the single InsecureRequestWarning from urllib3 needed when verify=False

urllib3.disable\_warnings(urllib3.exceptions.InsecureRequestWarning)

# Configure Tor proxy (for Tor Browser, default port is 9150)

proxies = {

'http': 'socks5h://127.0.0.1:9150',

'https': 'socks5h://127.0.0.1:9150'

}

headers = {

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0'

}

onion\_sites = [

'https://ahmia.fi',

'http://dreadytofatroptsdj6io7l3xptbet6onoyno2yv7jicoxknyazubrad.onion',

'http://torlib7fmhyvfv2k7s77xigdds3rosio6k6bxnn256xmtzlbgyizduqd.onion'

]

found\_onion\_links = set()

def extract\_onion\_links(html, base\_url):

soup = BeautifulSoup(html, 'html.parser')

links = soup.find\_all('a', href=True)

for link in links:

href = link['href']

match = re.search(r'(http[s]?://[a-zA-Z0-9]{10,56}\.onion)', href)

if match:

full\_url = match.group(1)

found\_onion\_links.add(full\_url)

def visit\_site(url):

print(f"[+] Visiting: {url}")

try:

# Disable SSL verification to avoid errors with self-signed certs on onion sites

response = requests.get(url, proxies=proxies, headers=headers, timeout=90, verify=False)

if response.status\_code == 200:

print(f" [✔] Success: {url}")

extract\_onion\_links(response.text, url)

else:

print(f" [!] Status Code {response.status\_code} for {url}")

except requests.exceptions.RequestException as e:

print(f" [ERROR] Could not access {url} → {e}")

def main():

print("[\*] Starting Dark Web Monitor via Tor")

for site in onion\_sites:

visit\_site(site)

time.sleep(5) # Delay to avoid rapid access and blocking

if found\_onion\_links:

print("\n[✔] Total onion links discovered:")

for link in found\_onion\_links:

print(f" - {link}")

else:

print("\n[!] No onion links discovered.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Key Points:**

proxies = {

'http': 'socks5h://127.0.0.1:9150',

'https': 'socks5h://127.0.0.1:9150'

}

Sets up **Tor proxy** so all requests go through Tor Browser (port 9150).

onion\_sites = ['https://ahmia.fi', 'http://dready...onion', 'http://torlib...onion']

List of websites to visit: Ahmia search engine + 2 dark web markets.

def extract\_onion\_links(html, base\_url):

soup = BeautifulSoup(html, 'html.parser')

links = soup.find\_all('a', href=True)

for link in links:

href = link['href']

match = re.search(r'(http[s]?://[a-zA-Z0-9]{10,56}\.onion)', href)

if match:

found\_onion\_links.add(match.group(1))

Parses HTML, looks for .onion links with regex, and adds them to a set.

def visit\_site(url):

response = requests.get(url, proxies=proxies, headers=headers, timeout=90)

if response.status\_code == 200:

extract\_onion\_links(response.text, url)

**Output:**

Prints all .onion links found.

No text or content analysis — just **link discovery**.