Name Mohsin Nawaz

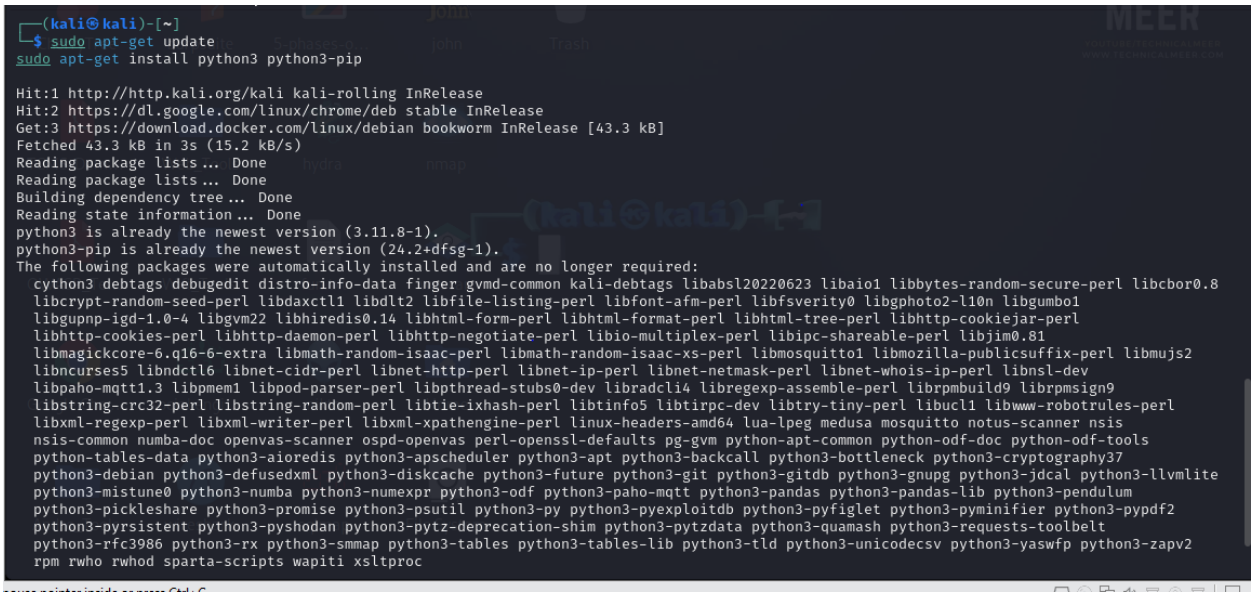
BrainWave Matrix Solutions

Cyber Security and Ethical Hacking Intern TASKS

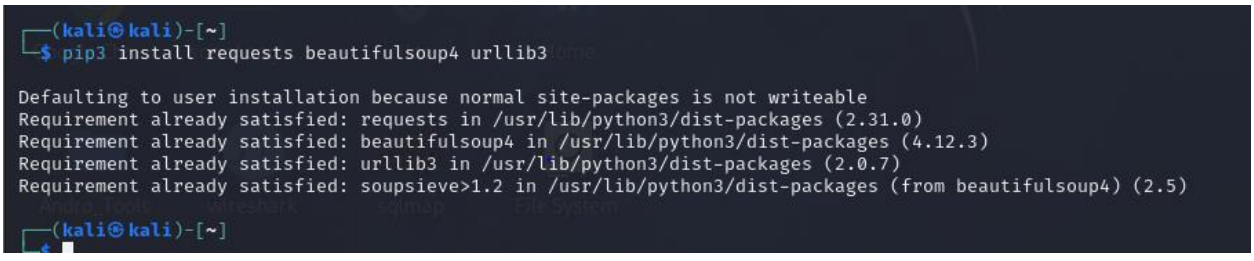
Question : Build a Python Cyber Security - Phishing Link Scanner

Answer

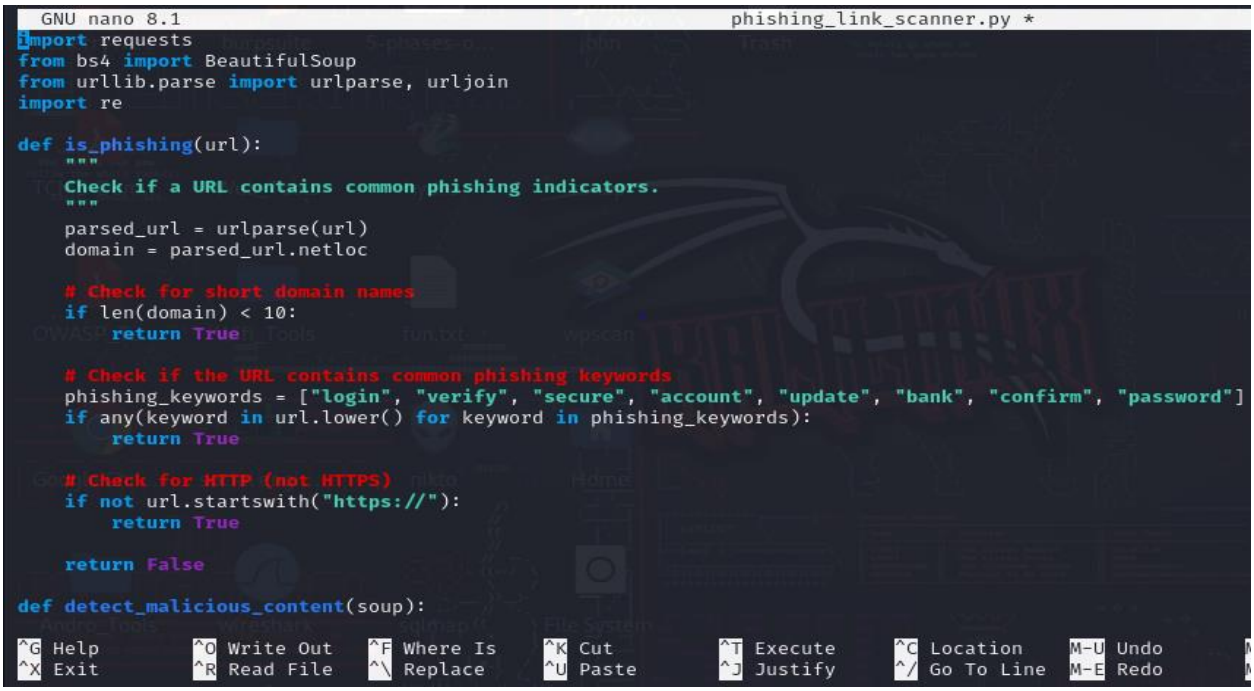
Installation of python files in kali linux terminal :



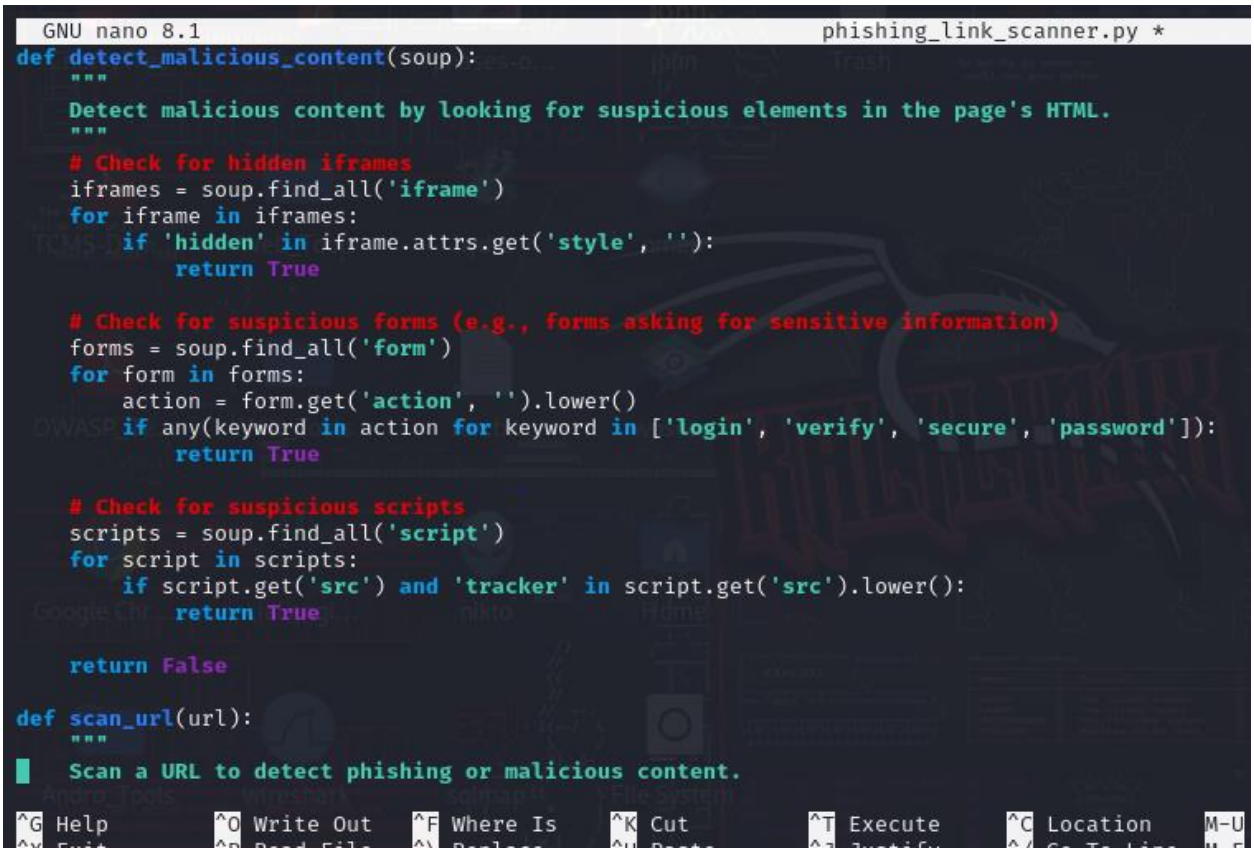
Installing the required libraries:



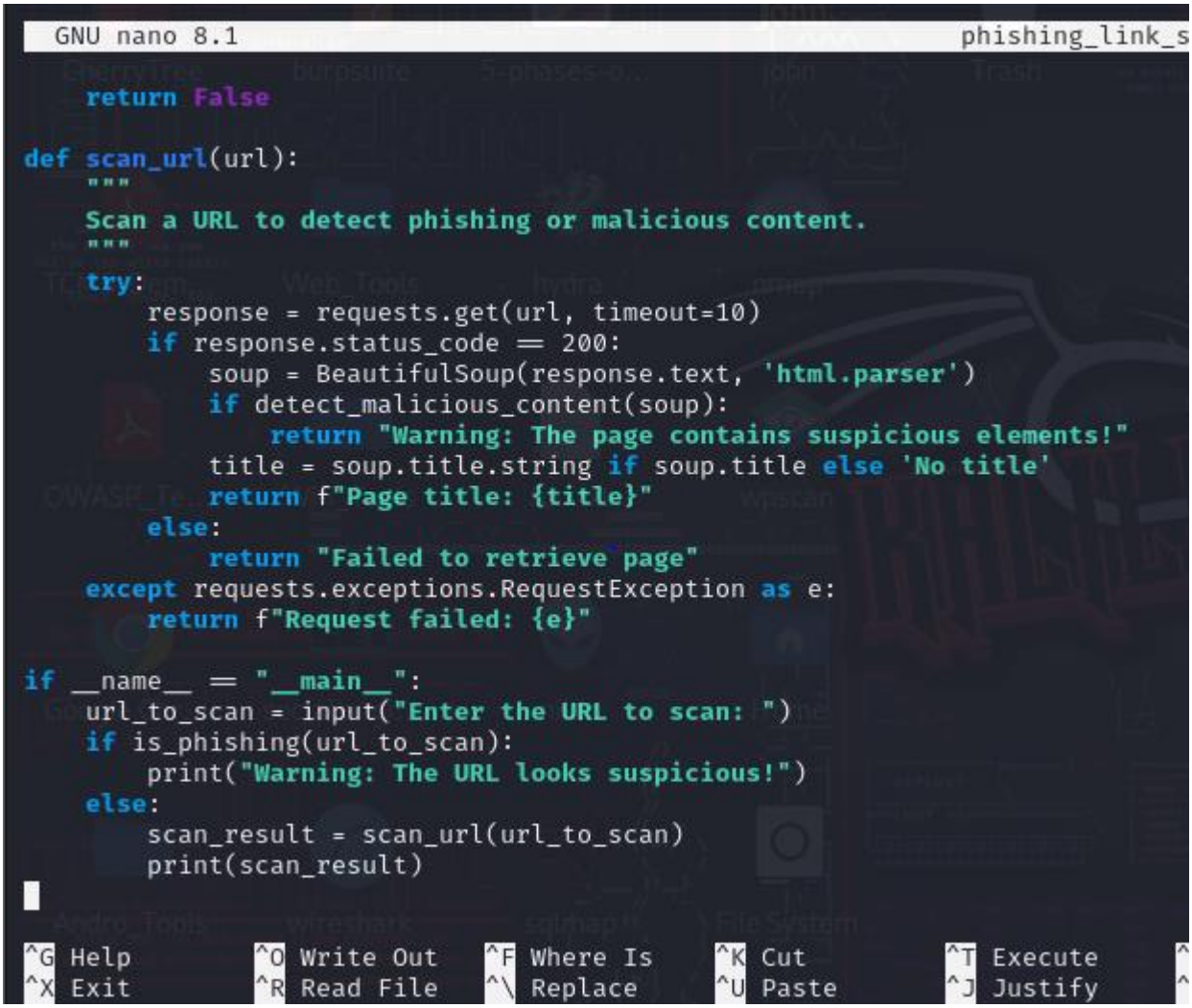
Create a Python script named `phishing_link_scanner.py` using a text editor like `nano`



```
  GNU nano 8.1                                  phishing_link_scanner.py *
import requests
from bs4 import BeautifulSoup
from urllib.parse import urlparse, urljoin
import re

def is_phishing(url):
    """
    Check if a URL contains common phishing indicators.
    """
    parsed_url = urlparse(url)
    domain = parsed_url.netloc

    # Check for short domain names
    if len(domain) < 10:
        return True

    # Check if the URL contains common phishing keywords
    phishing_keywords = ["login", "verify", "secure", "account", "update", "bank", "confirm", "password"]
    if any(keyword in url.lower() for keyword in phishing_keywords):
        return True

    # Check for HTTP (not HTTPS)
    if not url.startswith("https://"):
        return True

    return False

def detect_malicious_content(soup):
^G Help        ^O Write Out   ^F Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo
```



```
  GNU nano 8.1                                  phishing_link_scanner.py *
def detect_malicious_content(soup):
    """
    Detect malicious content by looking for suspicious elements in the page's HTML.
    """
    # Check for hidden iframes
    iframes = soup.find_all('iframe')
    for iframe in iframes:
        if 'hidden' in iframe.attrs.get('style', ''):
            return True

    # Check for suspicious forms (e.g., forms asking for sensitive information)
    forms = soup.find_all('form')
    for form in forms:
        action = form.get('action', '').lower()
        if any(keyword in action for keyword in ['login', 'verify', 'secure', 'password']):
            return True

    # Check for suspicious scripts
    scripts = soup.find_all('script')
    for script in scripts:
        if script.get('src') and 'tracker' in script.get('src').lower():
            return True

    return False

def scan_url(url):
    """
    Scan a URL to detect phishing or malicious content.
^G Help        ^O Write Out   ^F Where Is    ^K Cut         ^T Execute     ^C Location    M-U
```

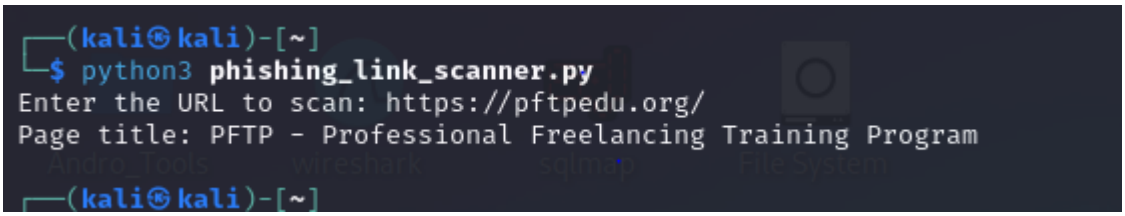


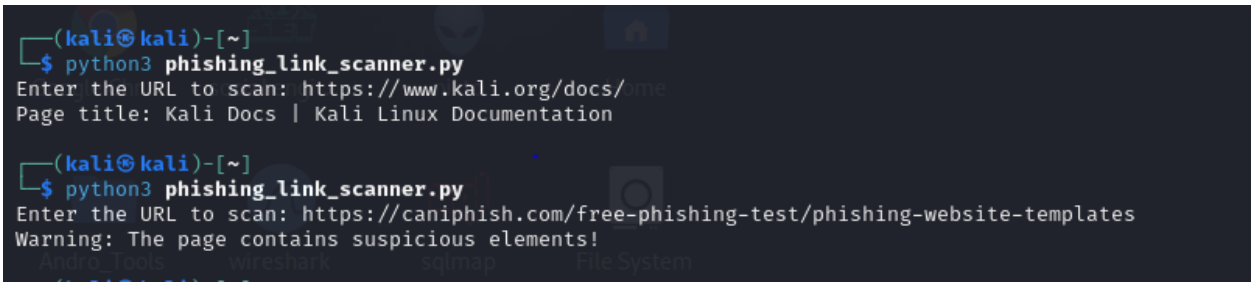
Explanation of the Code:

- **is_phishing Function**: This function checks for common phishing indicators like short domain names, common phishing keywords in the URL, and the use of HTTP instead of HTTPS.
- **detect_malicious_content Function**: This function checks the page's content for hidden iframes, suspicious forms, and scripts that might indicate malicious activity.
- **scan_url Function**: This function fetches the webpage content using requests, parses it with BeautifulSoup, and uses detect_malicious_content to identify any suspicious elements.
- **Main Functionality**: The script takes a URL as input, checks for phishing indicators, and scans the webpage for malicious content.

Make the script executable so

Now executing the python file so:





Enhancing the Phishing Link Scanner

Integrate with Threat Intelligence APIs

- Consider integrating APIs like VirusTotal or Google Safe Browsing to cross-check URLs against a known database of phishing and malicious domains.

Add Logging

- Implementing logging to record the results of each scan. Python's `logging` module can be used for this.

Implement Machine Learning

- Add a machine learning component to classify URLs based on a dataset of known phishing and non-phishing URLs for more accurate detection.

Security Considerations

- **Environment**: Ensure that to run the script in a secure environment since it fetches and processes content from external sites.
- **Ethical Use**: Always use the phishing link scanner ethically, ensuring the respect privacy and comply with legal guidelines.

This detailed implementation will provide us with a functional phishing link scanner in Kali Linux, capable of identifying potentially malicious URLs based on various indicators.

****