5. Scanning a website for vulnerabilities

This program could use the requests library in Python to send requests to a website and check for common vulnerabilities such as SQL injection and cross-site scripting (XSS).

Objective: To demonstrate how to scan a website for vulnerabilities using Python.

Tools Used: Python 3, requests library.

Steps:

1. Install the requests library using pip by running the following command in the terminal: **pip install requests**.
2. Write a Python program that sends requests to the target website and checks for common vulnerabilities such as SQL injection and cross-site scripting (XSS).

```python
import requests


# Define the target website URL
url = "https://www.example.com/"


# Send a GET request to the target website
response = requests.get(url)


# Check for SQL injection vulnerabilities
if "sql syntax" in response.text.lower():
    print("[!] SQL injection vulnerability detected on the target website!")
else:
    print("No SQL injection vulnerability detected.")


# Check for XSS vulnerabilities
if "<script>" in response.text.lower():
    print("[!] Cross-site scripting (XSS) vulnerability detected on the target website!")
else:
    print("No Cross-site scripting (XSS) vulnerability detected.")
```

3. Run the Python program in the terminal by typing **python filename.py**, where **filename.py** is the name of your Python file.

4. The program will send a GET request to the target website and check for SQL injection and XSS vulnerabilities.
5. If a vulnerability is detected, the program will print a warning message to the console.
6. You can modify the program to check for other vulnerabilities as well, depending on the target website and your requirements.