**Domain Phishing Detection Tool – Python Report**

**Objective**

The purpose of this project is to create a Python script that checks if a given domain name is a misspelled or suspicious version of any known, legitimate domain. This helps detect suspicious word changes, which is often used in phishing attacks to trick users into visiting fake websites that look similar to trusted ones.

**Overview**

The script uses Python’s built-in difflib module to compare the similarity between a user-entered domain and a list of known safe domains. The tool flags domains that are not exact matches but are highly similar, as these might be intentional misspellings used for malicious purposes.

**How It Works**

**1. String Similarity Check**

The core of the logic relies on a function called SequenceMatcher from the difflib module. This function returns a similarity score between 0 and 1. A score of 1 means the two strings are exactly the same, and lower values indicate more differences.

**2. Threshold Use**

A threshold value (default is 0.8) is used to decide how similar two domain names must be before the script flags one as suspicious. For example, if the input domain is g00gle.com, it has a similarity score of around 0.91 compared to google.com, which is above the threshold and therefore considered potentially dangerous.

**3. Exact Match Exception**

If the domain entered by the user is exactly the same as one of the legitimate domains, the script considers it safe and does not flag it, even if the similarity score is 1.

**Code Structure**

**1. misspelled\_domain() Function**

This function:

* Converts both the input domain and each legitimate domain to lowercase.
* Compares them directly for an exact match.
* If not an exact match, checks the similarity score.
* Returns True if the similarity is above the threshold, meaning the domain could be a typo-squatting attempt.

**2. main() Function**

This function:

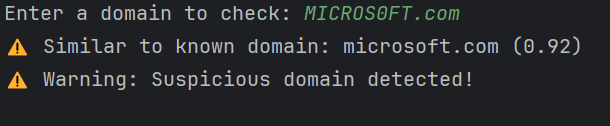
* Contains a predefined list of legitimate domains.
* Asks the user to enter a domain name using the input() function.
* Calls the misspelled\_domain() function with the user input.
* Prints whether the domain is safe or potentially suspicious.

**Sample Input and Output**

**Input:**

MICROS0FT.com

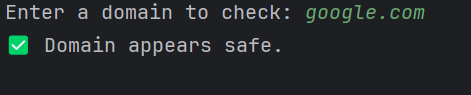
**Output:**



**Input:**

google.com

**Output:**



**Conclusion**

This Python tool helps detect suspicious domain names that closely resemble trusted websites. It works by comparing similarity scores and using a threshold to decide when to raise a warning. The code is written in a simple and readable way and avoids complex external libraries, making it ideal for beginners or use in a controlled environment like an educational project or demo.

The use of input() instead of command-line arguments improves user experience during development and testing.

**Optional Ideas for Extension**

* Allow checking multiple domains in one run.
* Load trusted domains from an external file.
* Let users adjust the similarity threshold.
* Save flagged results to a log file.