Обнаружение поддоменов определенного веб-сайта позволяет изучить всю его доменную инфраструктуру. Что это вообще такое? **Домен** — это адрес сайта в интернете, а поддомен — это домен, который является частью домена более высокого уровня. Для более лучшего понимания, например, у нас есть квартира, которую хозяин, после заселения, решил разделить на отдельные зоны. (для сна, отдыха, питания и так далее) И вся квартира будет выступать у нас в качестве домена, а эти функциональные зоны – поддомены. Если искать их вручную, то на это уйдет целая вечность. Но к счастью, можно написать программу на Python, которая будет делать это за нас, используя метод полного перебора, суть которого состоит в проверке всех общих имен поддоменов конкретного домена, всякий раз, когда мы получаем ответ от сервера, то есть - индикатор того, что поддомен жив. И в данной работе, я исследую сайт GitHub.

Первым делом, нам понадобится установить библиотеку requests, с помощью команды: pip3 install requests, после чего написать код, в котором мы ее импортируем и указываем домен для поиска поддоменов:

import requests

# домен для поиска поддоменов

domain = "github.com"

Далее, я использовал текстовый файл “subdomains-1000.txt”, содержащий список из 1000 поддоменов, наличие которых мы хотим проверить. Его я закинул в папку с моим проектом, после чего был написан следующий код:

# читать все поддомены

file = open("subdomains-1000.txt")

# прочитать весь контент

content = file.read()

# разделить на новые строки

subdomains = content.splitlines()

# список обнаруженных поддоменов

discovered\_subdomains = []

for subdomain in subdomains:

# создать URL

url = f"http://{subdomain}.{domain}"

try:

# если возникает ОШИБКА, значит, субдомен не существует

requests.get(url)

except requests.ConnectionError:

# если поддомена не существует, просто передать, ничего не выводить

pass

else:

print("[+] Обнаружен поддомен:", url)

# добавляем обнаруженный поддомен в наш список

discovered\_subdomains.append(url)

В нем мы создаем URL-адрес, подходящий для отправки запроса, затем мы используем функцию requests.get() для получения HTTP-ответа от сервера, что вызовает исключение ConnectionError всякий раз, когда сервер не отвечает, поэтому мы обернули это в блоке try/except. Если же исключение не возникло, значит субдомен существует. Запишем все обнаруженные поддомены в файл:

# сохраняем обнаруженные поддомены в файл

with open("discovered\_subdomains.txt", "w") as f:

for subdomain in discovered\_subdomains:

print(subdomain, file=f)

Как только процесс поиска будет завершен, вы увидите новый файл **discover\_subdomains.txt**, который включает все обнаруженные поддомены.