УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 4

по предмету «Компьютерные системы и сети»

Выполнил:

Горник А.

гр. 351003

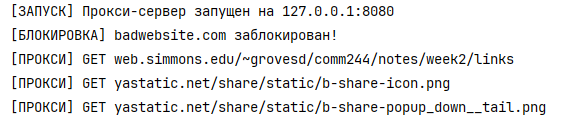
Проверил:

Леванцевич В.А.

Минск, 2025

*#http://web.simmons.edu/~grovesd/comm244/notes/week2/links  
#http://old-dos.ru/index.php?page=files&mode=files&do=show&id=100953  
#http://badwebsite.com/*import socket  
import threading  
  
PROXY\_HOST = '127.0.0.1'  
PROXY\_PORT = 8080  
  
*# Черный список доменов (для блокировки)*BLACKLIST = {"example.com", "badwebsite.com"}  
  
  
def handle\_client(client\_socket):  
 try:  
 request = client\_socket.recv(4096)  
 headers = request.decode(errors='ignore').split("\r\n")  
 request\_line = headers[0]  
 method, full\_url, protocol = request\_line.split()  
  
 *# Обработка абсолютного URL (как у браузеров через прокси)* if full\_url.startswith("http://")==False:  
 response = "HTTP/1.1 400 Bad Request"  
 client\_socket.send(response.encode())  
 client\_socket.close()  
 return  
  
 url = full\_url[len("http://"):]  
 host = url.split("/")[0]  
 path = "/" + "/".join(url.split("/")[1:])  
  
 *# Проверка чёрного списка* if host in BLACKLIST:  
 response = "HTTP/1.1 403 Forbidde"  
 client\_socket.send(response.encode())  
 print(f"[БЛОКИРОВКА] {host} заблокирован!")  
 client\_socket.close()  
 return  
  
 *# Переписываем request-line (с абсолютного URL на относительный путь)* new\_request\_line = f"{method} {path} {protocol}"  
 request = request.replace(request\_line.encode(), new\_request\_line.encode())  
  
 *# Подключаемся к целевому серверу* server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 server\_socket.connect((host, 80))  
 server\_socket.sendall(request)  
  
 while True:  
 response = server\_socket.recv(4096)  
 if not response:  
 break  
 client\_socket.send(response)  
  
 print(f"[ПРОКСИ] {method} {host}{path}")  
  
 except Exception:  
 pass  
 finally:  
 client\_socket.close()  
  
def start\_proxy():  
 proxy\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 proxy\_socket.bind((PROXY\_HOST, PROXY\_PORT))  
 proxy\_socket.listen(1)  
 print(f"[ЗАПУСК] Прокси-сервер запущен на {PROXY\_HOST}:{PROXY\_PORT}")  
  
 while True:  
 client\_socket, \_ = proxy\_socket.accept()  
 client\_thread = threading.Thread(target=handle\_client, args=(client\_socket,))  
 client\_thread.start()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 start\_proxy()

**Тесты:**

****