Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

**БЕЗОПАСНОСТЬ СЕТЕЙ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН**

Отчет по выполнению лабораторной работы №6

Выполнил

ст. гр. 230781 А. Р. Ивлев

Проверил

Асс. Греков Михаил Михайлович

Тула 2022

**Лабораторная работа №6**

**ФОРМИРОВАНИЕ И ПЕРЕДАЧ ПО СЕТИ ПАКЕТОВ СТЕКА TCP/IP С ИСПОЛЬЗОВАНИЕМ WINPCAP**

**Цель работы**

Научиться формировать пакеты стека протоколов TCP/IP и передавать их по сети, используя архитектуру WinPCAP.

**Задание на работу**

*Вариант №1*

Разработать программу, которая формирует TCP пакеты и отправляет их в сеть. В программе предусмотреть диалоговый ввод IP – адреса и порта получателя.

**Ход работы**

Был написана программа на языке Python, которая отправляет TCP пакет определенному получателю.

*Листинг 1.* *Анализатор TCP пакетов*

from winpcapy import WinPcap, WinPcapUtils

def send\_packet\_device\_name(device\_name, packet\_buffer):

with WinPcap(device\_name) as capture:

capture.send(packet\_buffer)

def ethernetip(src\_mac=b'\x00\x00\x00\x00\x00\x00', dst\_mac=b'\x00\x00\x00\x00\x00\x00'):

type = b'\x08\x00'

header = dst\_mac + src\_mac + type

return header

def ipheader(src\_ip, dst\_ip):

version = b'\x45\x00'

length = b'\x00\x28'

identification = b'\xab\xcd'

flags = b'\x00\x00'

protocol = b'\x40\x06'

checksum = b'\x00\x00'

header = version + length + identification + flags + protocol + checksum + src\_ip + dst\_ip

return header

def tcpheader(src\_port, dst\_port):

sequence = b'\x00\x00\x00\x00'

ack = b'\x00\x00\x00\x00'

flags = b'\x50\x02\x00\x00'

window = b'\xff\xff'

checksum = b'\x00\x00'

urgent = b'\x00\x00'

header = src\_port + dst\_port + sequence + ack + flags + window + checksum + urgent

return header

def send\_packet(device\_name, src\_ip, src\_port, dst\_ip, dst\_port, msg):

src\_ip = b''.join([int(i).to\_bytes(1, 'big') for i in src\_ip.split('.')])

dst\_ip = b''.join([int(i).to\_bytes(1, 'big') for i in dst\_ip.split('.')])

src\_port = src\_port.to\_bytes(2, 'big')

dst\_port = dst\_port.to\_bytes(2, 'big')

msg = msg.encode('utf-8')

ethernet = ethernetip()

ip\_header = ipheader(src\_ip, dst\_ip)

tcp\_header = tcpheader(src\_port, dst\_port)

packet = ethernet + ip\_header + tcp\_header + msg

send\_packet\_device\_name(device\_name, packet)

device\_name = "\\Device\\NPF\_{EE5AAD83-CD76-4A0B-8C83-C120882BE16A}"

src\_ip = '192.168.136.1'

src\_port = 12345

dst\_ip = input(f"{'Destination IP:': <20}")

dst\_port = int(input(f"{'Destination port:': <20}"))

msg = input(f"{'Message:': <20}")

send\_packet(device\_name, src\_ip, src\_port, dst\_ip, dst\_port, msg)

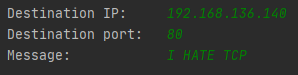


Рисунок 1 – Результат работы скрипта

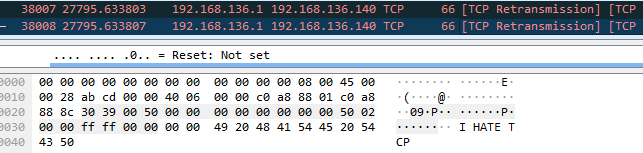


Рисунок 2 – Отправленный TCP/IP пакет

**Вывод**

Была разработана программа, которая с помощью WINPCAP формирует TCP пакеты и отправляет их в сеть. В программе предусмотрен диалоговый ввод IP – адреса и порта получателя.

WinPcap – стандартный инструмент для доступа к сети на канальном уровне в среде Windows, который позволяет приложениям захватывать и передавать сетевые пакеты в обход стека протоколов.

WinPcap состоит из драйвера, который расширяет операционную систему для обеспечения низкоуровневого доступа к сети, и библиотеки, которая используется для легкого доступа к низкоуровневым сетевым уровням. Эта библиотека также содержит Windows-версию известного Unix API libpcap.