1. Министерство науки и высшего образования Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Высшая школа кибербезопасности

**ЛАБОРАТОРНАЯ РАБОТА № 6**

1. **«Исследование Бот-сетей»**
2. по дисциплине «Программно-аппаратные средства обеспечения информационной безопасности»
3. Выполнил
4. студент гр. 5151004/10101 Плохотникова М.С.

<*подпись*>

1. Проверил
2. преподаватель Семенов П.О.

<*подпись*>

1. Санкт-Петербург
2. 2024

# Цель работы

Написать программу, которая путем анализа параметров запросов и ответов DNS-сервера определит инфицированные машины ЛВС №1.

# ход решения

По условию лабораторной работы дана сеть, разделенная на ЛВС №1, ЛВС №2 и DNS-сервер. ЛВС №1 содержит компьютеры с адресами: 192.168.50.10 – 192.168.50.35, некоторые из которых инфицированы и являются ботами. ЛВС №2 (192.168.50.50 – 192.168.50.70) содержит защищенные компьютеры, на которых функционируют легитимные приложения.

В выданном варианте содержится PCAP-файл с записями запросов к веб-серверу и его ответов.

Для обнаружения подозрительных пакетов был написан скрипт на Python.

Обработка пакетов осуществляется средствами библиотеки scapy.

Сначала проверяется, что пакет является DNS-запросом к серверу 192.168.50.88 и адрес источника из ЛВС №1. Если данное условие выполняется, то идет проверка состояния полей заголовка DNS.

Для DNS-запросов характерны следующие значения заголовка:

* Opcode установлен в значение 0-2, так как остальные зарезервированы и не используются.
* Флаг z зарезервирован, и поэтому всегда равен 0.
* Флаги ancount, nscount и arcount равны 0.

Если полученный пакет, имеет отличные значения заголовка, то он заносится в список подозрительных.

В результате работы программы был обнаружен один подозрительный источник DNS-запросов с некорректными полями DNS-заголовка.

Также был осуществлен поиск подозрительных пакетов по запрашиваемым доменам.

Результат работы программы приведен на рисунке ниже.

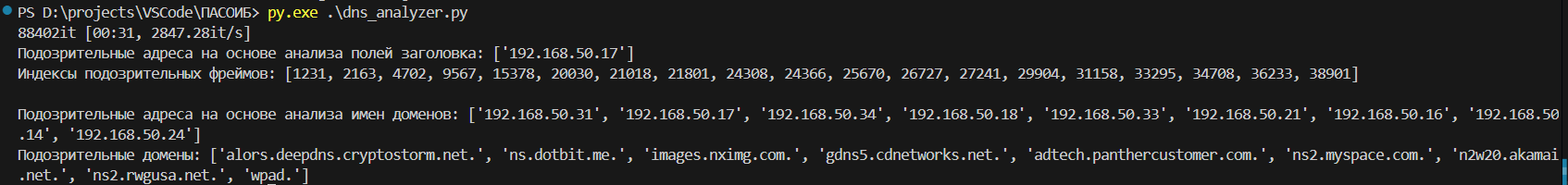


Рисунок – Результат работы программы

# Выводы

DNS-туннелирование позволяет злоумышленникам скрывать вредоносный трафик в обычном DNS-трафике. Стандартные средства безопасности, такие как брандмауэры, могут не обнаруживать и не блокировать DNS-туннелирование. Это делает DNS-туннелирование эффективным методом для обхода мер безопасности и передачи вредоносных данных.

ПРИЛОЖЕНИЕ А

Листинг программы:

from scapy.all import \*

from scapy.layers.inet import IP

from scapy.layers.dns import DNS

import ipaddress

from tqdm import tqdm

legitimate\_apps = [

"google.com.",

"dropbox.com.",

"utorrent.com.",

"bittorrent.com.",

"avast.com.",

"wps.com.",

"adobe.com.",

"browser-chrome.ru",

"firefox.com.",

"malwarebyte.com.",

"fosshub.com",

"audacityteam.org.",

"bytefence.com.",

"thunderbird.net.",

"mozila.org.",

"skype.com.",

"facebook.com.vn.",

"ccleaner.com.",

"dns.msftncsi.com.",

"windowsupdate.com.",

"windows.com."

]

pkt = PcapReader("var\_4.pcap")

end\_ip = ipaddress.ip\_address("192.168.50.35")

suspicious\_adresses\_flags = []

suspicious\_adresses\_domain = []

suspicious\_domains = []

frame\_index = []

for i, p in enumerate(tqdm(pkt)):

if DNS not in p:

continue

if p[DNS].qr == 0 and p[IP].dst == "192.168.50.88":

if ipaddress.ip\_address(p[IP].src) < end\_ip:

domain = p[DNS].qd.qname.decode()

if p[DNS].opcode > 2 or p[DNS].z or p[DNS].ancount or p[DNS].nscount or p[DNS].arcount:

frame\_index.append(i+1)

if not p[IP].src in suspicious\_adresses\_flags:

suspicious\_adresses\_flags.append(p[IP].src)

elif not any(domain.endswith(app) for app in legitimate\_apps):

if not p[IP].src in suspicious\_adresses\_domain:

suspicious\_adresses\_domain.append(p[IP].src)

suspicious\_domains.append(domain)

print("Подозрительные адреса на основе анализа полей заголовка:", suspicious\_adresses\_flags)

print("Индексы подозрительных фреймов:", frame\_index)

print()

print("Подозрительные адреса на основе анализа имен доменов:", suspicious\_adresses\_domain)

print("Подозрительные домены:", suspicious\_domains)