

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ компьютерной безопасности и криптографии

**ОБНАРУЖЕНИЕ СЕТЕВОГО RDP ТРАФИКА МЕТОДОМ АНАЛИЗА
ЕГО ПОВЕДЕНИЯ**

КУРСОВАЯ РАБОТА

студента 3 курса 331 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Токарева Никиты Сергеевича

Научный руководитель
доцент

Гортинский А. В.

Заведующий кафедрой

Абросимов М. Б.

Саратов 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Протокол RDP	4
1.1 Принцип работы протокола RDP	4
1.2 Безопасность протокола RDP	5
2 Обнаружение сеанса удаленного управления	6
2.1 Клавиатурный мониторинг	6
ЗАКЛЮЧЕНИЕ	9
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	10
Приложение А Код keylogger-win.py	11
Приложение Б Код analyze-data.py	12

ВВЕДЕНИЕ

Информация – это сведения об окружающем мире и протекающих в нём процессах, которые зафиксированы на каком-либо носителе. Благодаря протоколам удаленного доступа можно распоряжаться базами данных, информацией, которая хранится на другом устройстве. В недавнем прошлом большинство схем удаленного доступа характеризовалось высокой стоимостью, низкой производительностью, небольшой скоростью передачи данных, недостаточным уровнем защищенности передаваемой информации [1].

1 Протокол RDP

Протокол RDP (от англ. Remote Desktop Protocol — протокол удалённого рабочего стола) — патентованный протокол прикладного уровня компании Microsoft и приобретен ею у другой компании Polycom, который предоставляет пользователю графический интерфейс для подключения к другому компьютеру через сетевое соединение. Для этого пользователь запускает клиентское программное обеспечение RDP, а на другом компьютере должно быть запущено программное обеспечение сервера RDP [2].

Клиенты для подключения по RDP существуют для большинства версий Microsoft Windows, Linux, Unix, macOS, iOS, Android и других операционных систем. Стоит отметить, что RDP-серверы встроены в операционные системы Windows. По умолчанию подключения, созданные с помощью RDP, используют TCP-порт 3389, по которому осуществляется передача данных.

1.1 Принцип работы протокола RDP

Принцип работы RDP базируется на протоколе TCP. Соединение клиент-сервер происходит на транспортном уровне. После инициализации пользователь проходит аутентификацию. В случае успешного подтверждения сервер передает клиенту управление.

Протокол RDP внутри себя поддерживает виртуальные каналы, через которые пользователю передаются дополнительные функции операционной системы, например, можно распечатать документ, воспроизвести видео или скопировать файл в буфер обмена.

Известно, что RDP является прикладным протоколом, базирующимся на TCP. Для начала пользователю необходимо установить соединение клиент-сервер, которое происходит на транспортном уровне. После инициализации RDP-сессии производится аутентификация. Далее сервер начинает передавать клиенту графический вывод и ожидает входные данные от клавиатуры и мыши. В качестве графического вывода может выступать как точная копия графического экрана, передаваемая как изображение, так и команды на отрисовку графических примитивов, например, линия, круг, эллипс, текст и др. Для протокола RDP приоритетом является передача вывода с помощью примитивов, так как это экономит трафик. Изображение передается только в том случае, если не удалось согласовать параметры передачи примитивов при установке RDP-

сессии. Обработка полученных команд и вывод изображения осуществляется с помощью графической подсистемы RDP-клинта. Сигнал нажатия и отпускания клавиши клавиатуры шифруются и ожидают команды отправки [3].

1.2 Безопасность протокола RDP

Как уже известно, что для операционной системы Windows постоянно выходят различные обновления, включая обновлений RDS (от англ. Remote Desktop Services — службы удаленных рабочих столов). В связи с этим возникают различные уязвимости при инициализации RDP-сессии. В основном они не связаны непосредственно с протоколом RDP, но касаются службы удаленных рабочих столов RDS и позволяют при успешной эксплуатации путем отправления специального запроса через RDP получить возможность выполнения произвольного кода на уязвимой системе, даже не проходя при этом процедуру проверки подлинности. Достаточно лишь иметь доступ к хосту или серверу с уязвимой системой Windows. Таким образом, любая система, доступная из сети Интернет, является уязвимой при отсутствии установленных последних обновлений безопасности Windows.

Если стоит задача защитить удаленный доступ, то, конечно, необходимо использовать надежный пароль, обновить свое программное обеспечение до последней версии, также можно использовать VPN подключение, чтобы получить IP-адрес виртуальной сети и добавить его в правило исключения брандмауэра RDP. Стоит отметить, что существует много разных способов, чтобы защитить подключение с помощью протокола RDP и более подробно это описано в документации Microsoft.

2 Обнаружение сеанса удаленного управления

2.1 Клавиатурный мониторинг

Для начала создается сеанс удаленного управления. Допустим *устройство 1* подключилось к *устройству 2*. На *устройстве 2* запускается программа keylogger-win.ру, отслеживающая клавиши и время их нажатия. Далее *пользователь 1* вводит небольшой текст. При закрытии консоли или нажатии комбинации клавиш *Ctrl + f5*, программа завершает свою работу и в итоге получается log-файл, как показано на рисунке 1.

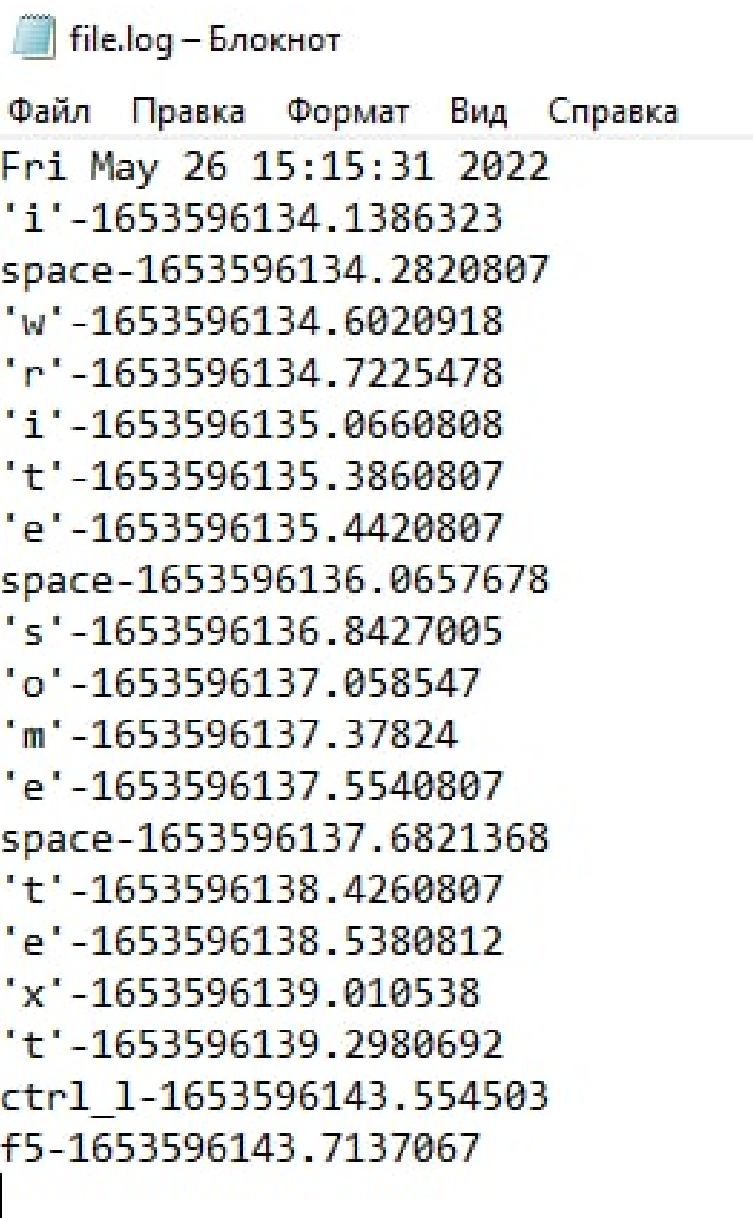
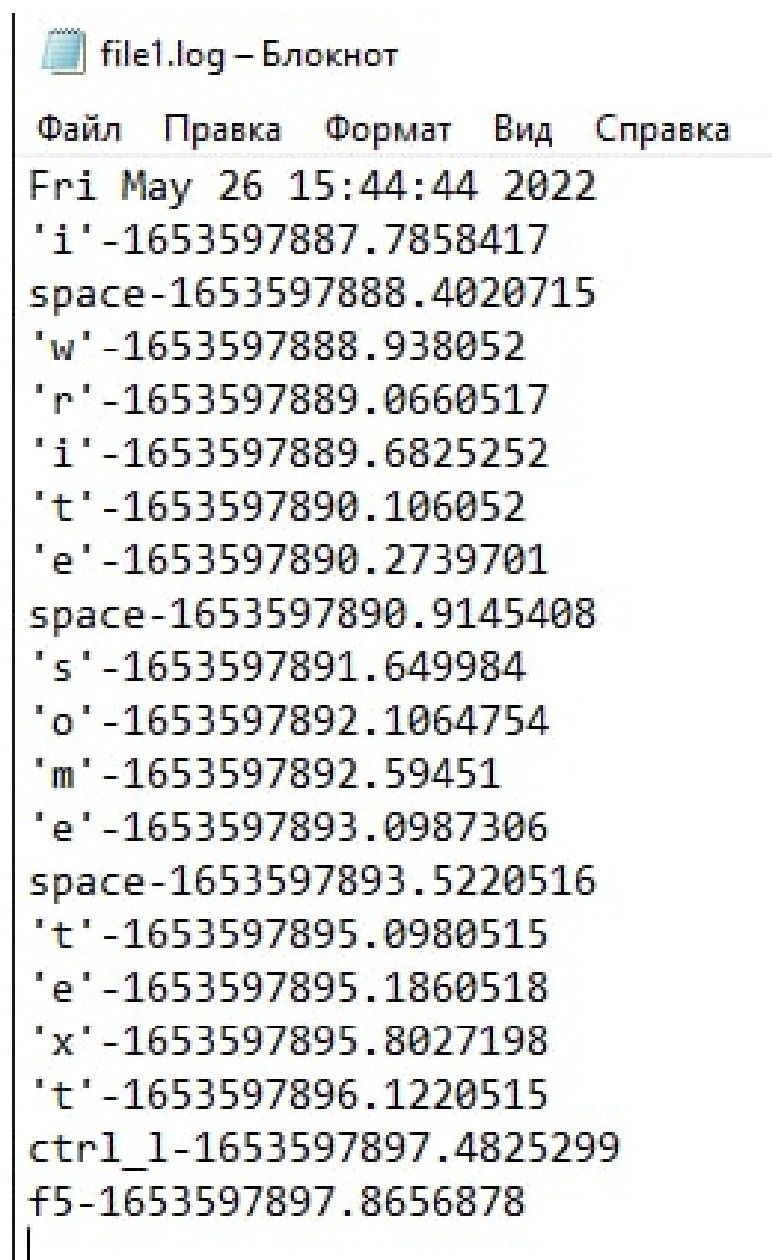


Рисунок 1 – Содержимое файла file.log

Далее опять запускается данная программа, но уже тот же самый текст набирает *пользователь 2*. И в итоге получается новый файл file1.log, как показано

на рисунке 2.



```
file1.log – Блокнот
Файл  Правка  Формат  Вид  Справка
Fri May 26 15:44:44 2022
'i' -1653597887.7858417
space-1653597888.4020715
'w' -1653597888.938052
'r' -1653597889.0660517
'i' -1653597889.6825252
't' -1653597890.106052
'e' -1653597890.2739701
space-1653597890.9145408
's' -1653597891.649984
'o' -1653597892.1064754
'm' -1653597892.59451
'e' -1653597893.0987306
space-1653597893.5220516
't' -1653597895.0980515
'e' -1653597895.1860518
'x' -1653597895.8027198
't' -1653597896.1220515
ctrl_1-1653597897.4825299
f5-1653597897.8656878
```

Рисунок 2 – Содержимое файла file1.log

Затем эти файлы отправляются на анализ для построения гистограммы. На рисунке 3 изображен ввод данных, где программа запрашивает у пользователя количество файлов для анализа, их имена, а также дает возможность построить гистограмму для любых двух заданных файлов.

```
Введите количество файлов для анализа
2
Введите название 1-го файла (формат <имя_файла>.log):
file.log
Введите название 2-го файла (формат <имя_файла>.log):
file1.log
Построить гистограмму? (Нажмите "1"))
1
Выберите два файла для построения (1-2)
1 2
Построить гистограмму? (Нажмите "1"))
```

Рисунок 3 – Ввод данных для анализа

На рисунке 4 показана диаграмма, в которой можно увидеть разницу пауз нажатия клавиш.

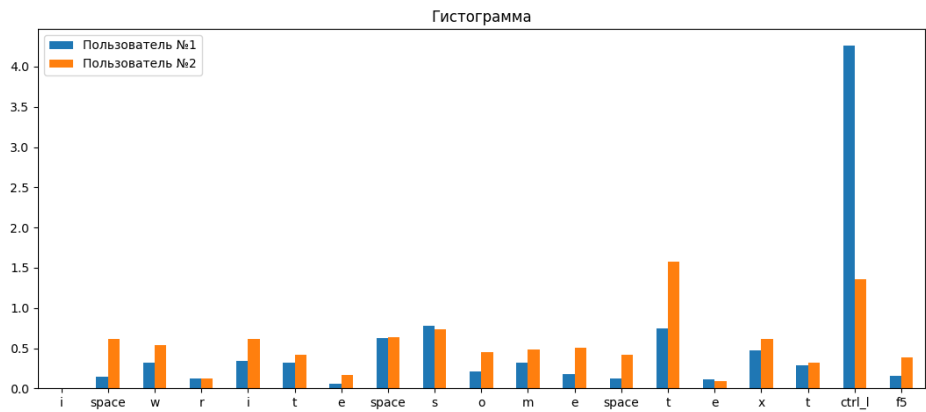


Рисунок 4 – Гистограмма

Можно сделать вывод, что паузы нажатия клавиш, между *пользователем 1* и *пользователем 2* различаются. Но всё таки эту разницу возможно свести к минимуму, если *пользователь 2* постарается подделать почерк *пользователя 1*. Поэтому такой подход будет считаться ненадежным и бессмысленным.

ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен протокол RDP. . .

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Книга Ибе О.С. «Компьютерные сети и службы удаленного доступа» / пер. с англ. - Москва, издательство: «ДМК Пресс», Яз. рус.
- 2 Удалённый рабочий стол RDP: как включить и как подключиться по RDP [Электронный ресурс] / URL:<https://hackware.ru/?p=11835> (дата обращения 03.05.2022), Яз. рус.
- 3 Документация по устранению неполадок служб удаленных рабочих стола для Windows Server [Электронный ресурс] / URL: <https://inlnk.ru/bvOV0> (дата обращения 04.05. 2022), Яз. рус.

ПРИЛОЖЕНИЕ А

Код keylogger-win.py

```
import pynput
import time
from pynput.keyboard import Key, Listener

is_ctrl = False
path = 'e:/file.log'

def press_elem(event):
    event = str(event)
    check = event.find('Key')
    if check != -1:
        event = event.replace('Key.', '')
    with open(path, 'a+') as f:
        f.write('{}-'.format(event))
        f.write('{}\n'.format(time.time()))

def release_elem(event):
    global is_ctrl
    if event == Key.ctrl_l:
        is_ctrl = True
    if event != Key.ctrl_l and event != Key.f5:
        is_ctrl = False
    if event == Key.f5 and is_ctrl:
        return False

if __name__ == '__main__':
    with open(path, 'a+') as f:
        f.write('{}\n'.format(time.ctime()))
    with Listener(on_press = press_elem, on_release = release_elem) as listener:
        listener.join()
```

ПРИЛОЖЕНИЕ Б

Код analyze-data.py

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

def get_data(file_name):
    file_ = open(file_name, 'r')
    elems = []
    time = []
    prev = -1
    fl = False
    while True:
        line = file_.readline()
        if not line:
            break
        sep = line.find('-')
        if sep != -1:
            elems.append(line[:sep].strip(""))
            time.append(float(line[sep + 1:]))
            prev += 1
        elif fl == False:
            time.append(0.0)
            prev += 1
            fl = True
        elif elems != []:
            break

    for i in range(1, len(time) - 1):
        time[i] = time[i + 1] - time[i]
    return elems, time[:len(time) - 1]

def create_diagram(events, data_time, frst, scnd):
    df = pd.DataFrame({frst : data_time[frst], scnd : data_time[scnd]}, index=events)
    ax = df.plot.bar(rot = 0)

    plt.title("Гистограмма")
    plt.show()

def mode():
    print('Построить гистограмму? (Нажмите "1")')
    bl = input()
    if bl == '1':
        name = 'Пользователь №'
        print(f'Выберите два файла для построения (1-{n})')
        s = input()
        frst, scnd = s[:s.find(' ')], s[s.find(' ') + 1:]
        create_diagram(data_events[name + frst], data_time, name + frst, name + scnd)
```

```

        return True
    else:
        return False

if __name__ == '__main__':
    print('Введите количество файлов для анализа')
    n = int(input())
    data_events = {}
    data_time = {}
    name = 'Пользователь №'
    for i in range(n):
        print(f'Введите название {i + 1}-го файла (формат <имя_файла>.log):')
        file_name = input()
        data_events[name + str(i + 1)], data_time[name + str(i + 1)] = get_data(file_name)
    fl = True
    while fl:
        fl = mode()

```