

**UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI**  
**Information and Communication Technology Department**



# **Final Report Part 1.1**

## **Malware Analyst**

*Student Name: Đào Ngọc Tùng*

*Student ID: BA12-185*

**Ha Noi, 12 December 2024**

## **Table Of Content**

### **1. Basic Static Analysis**

- a. Purpose .....**
- b. File Information .....**
- c. Virus Total .....**
- d. PEiD .....**
- e. Dependency Walker .....**
- f. Bintext .....**
- g. Conclusion .....**

### **2. Basic Dynamic Analysis**

- a. Purpose .....**
- b. Process Monitor .....**
- c. Process Explorer .....**
- d. Conclusion .....**

### **3. Advanced Static Analysis**

- a. Purpose .....**
- b. IDA .....**
- c. Conclusion .....**

### **4. Advanced Dynamic Analysis**

- a. OllyDbg .....**
- b. Conclusion .....**

# Basic Static Analysis

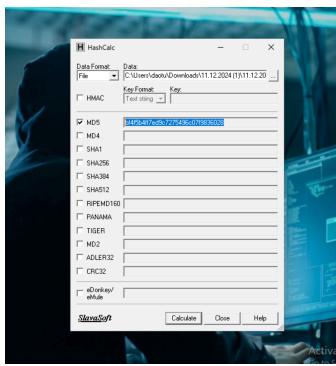
## Purpose

The idea here is to glean any information from the file itself. This information is useful because it gives us an idea of what to expect when we run the executable, and may give us enough information to tweak our tools to get more out of the following basic dynamic analysis.

## File information

File:part11.exe

MD5: bf4f5b4ff7ed9c7275496c07f9836028. Using HashClac to find MD5



Size File: 37KB.

	part11	9/18/2015 7:37 AM	Application	37 KB
--	--------	-------------------	-------------	-------

## VirusTotal

Detection ratio: 59/68

Packers Identified: Not packed

Creation Time: 2010-03-30 11:49:58 UTC

A screenshot of the VirusTotal analysis interface. The file 'part11.exe' has a community score of 59/68. The analysis was completed on 9/18/2015 at 7:37 AM. The file size is 36.50 KB. The report shows various detection details for different engines, including F-Secure, McAfee, and Kaspersky, among others. It also includes sections for file properties, history, and a detailed breakdown of each detection.

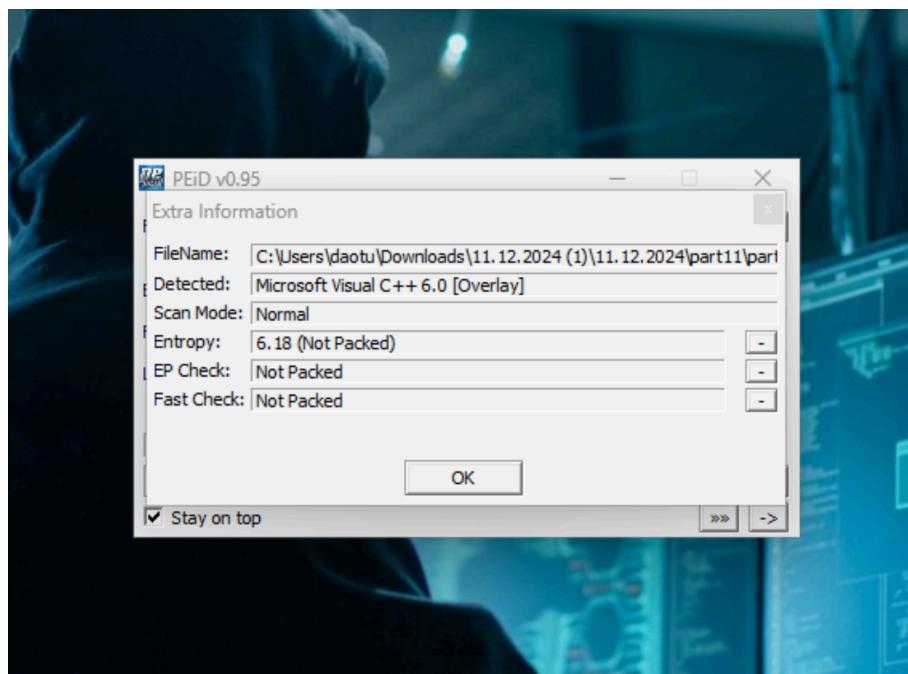
## Interesting Sections:

Sections						
Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	29114	29184	6.6	fec4f9ce4bdabdd0a69037ce73cc7bdd	169673.19
.rdata	36864	3446	3584	5.22	a1496b151ceb8edaa8b2ede718108456	85707.62
.data	40960	8988	3072	2.2	efbbde8ca7f252ba3dbcf87d5c04fe50	469798

Figure 1: Section information from VirusTotal

## PEiD

PEiD shows the file is not packed.



## Dependency Walker

KERNEL32.DLL

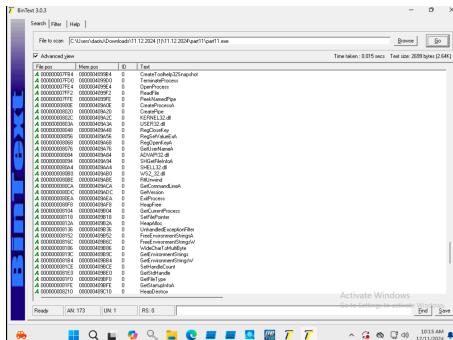
ADVAPI32.DLL

SHELL32.DLL

WS2\_32.DLL

A	000000000802C	000000409A2C	0	KERNEL32.dll
A	000000000803A	000000409A3A	0	USER32.dll
A	0000000008048	000000409A48	0	RegCloseKey
A	0000000008056	000000409A56	0	RegSetValueExA
A	0000000008068	000000409A68	0	RegOpenKeyA
A	0000000008076	000000409A76	0	GetUserNameA
A	0000000008084	000000409A84	0	ADVAPI32.dll
A	0000000008094	000000409A94	0	SHGetFileInfoA
A	00000000080A4	000000409AA4	0	SHELL32.dll
A	00000000080B0	000000409AB0	0	WS2_32.dll

## Bintext



Some function-like names that aren't in the imports list.

+)**KERNEL32.DLL**

- CloseHandle
- CopyFileA
- CreateFileA
- CreateFileMappingA
- CreatePipe
- CreateProcessA
- CreateToolhelp32Snapshot
- DeleteFileA
- ExitProcess
- FileTimeToSystemTime

+)**ADVAPI32.DLL**

- GetUserNameA
- RegCloseKey
- RegOpenKeyA
- RegSetValueExA

+)**SHELL32.DLL**

- SHGetFileInfoA

+)**WS2\_32.DLL**

- closesocket
- connect
- gethostbyname
- gethostname
- htonl
- inet\_addr
- inet\_ntoa
- recv
- send
- Socket

## Conclusion

The Total virus report has enough basic information for static analysis because the information is not packed.

## Basic Dynamic Analysis

### Purpose

By running the program and analyzing its effects on the system we can get a first-hand look of the malware in action. This gives us some context when we disassemble the executable during advanced static analysis. It also allows us to develop host or network based signatures to identify the presence of this malware on a system in the future.

### ProcessMonitor

Access crypto libs a lot, probably for the unpacking procedure.

Time ...	Process Name	PID	Operation	Path
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57902 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57903 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57903 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57903 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57904 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57904 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57904 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57905 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57905 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57905 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57906 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57906 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57906 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57907 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57907 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57907 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57908 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57908 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57908 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57909 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57909 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Receive	ngocung.localdomain:57909 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Connect	ngocung.localdomain:57910 -> lb-212-231.above.com:https
10:40....	part11.exe	2372	TCP Send	ngocung.localdomain:57910 -> lb-212-231.above.com:https

Showing 283 of 950,938 events (0.029%)

Backed by virtual memory

The file contains **4 types of operations (Operation)**. Here's the list of operation types and their general purposes:

#### 1. TCP Receive

- Description: Represents the activity of receiving data over a TCP network connection.
- Purpose: Used to gather data sent from a server or another device through a TCP link.

#### 2. TCP Connect

- Description: Represents the establishment of a new TCP connection between two devices (e.g., client and server).
- Purpose: Allows a process to connect to a server or another service over the network.

#### 3. TCP Send

- Description: Represents sending data over a TCP connection to the network.
- Purpose: Facilitates data transmission from the current process to a server or another device.

#### 7. TCP Reconnect

- Description: Represents re-establishing a TCP connection, usually after a previous connection was interrupted.
- Purpose: Maintains communication with a server or another device following a lost connection.

## Process Explorer

1. TCP/IP maintains multiple **CLOSE\_WAIT** TCP connections to **103.224.212.231** on port 443, suggesting improper connection handling. This may indicate inefficient coding or potential malicious activity; further investigation is needed.

The screenshot shows the 'TCP' tab in the Process Explorer properties window for process part11.exe:2372. A checkbox labeled 'Resolve addresses' is checked. The table lists 20 TCP connections, all of which are in the 'CLOSE\_WAIT' state. The local address for all connections is 'ngocung.localdomain....' and the remote address is '103.224.212.231:https'. The ports for these connections range from 103 to 123.

P...	Local Address	Remote Address	State
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT
TCP	ngocung.localdomain....	103.224.212.231:https	CLOSE_WAIT

2. Performance for the process, detailing resource usage. Key points include:

- CPU: Low usage with a total time of 5.9 seconds and priority 8.
- Memory: Moderate use of virtual memory (178,968 K virtual size) with 16,168 K in the working set.
- I/O: 16,531 "Other" I/O operations suggest unusual activity.
- Handles: 1,964 handles, which is relatively high.

This behavior could indicate potentially abnormal or resource-intensive activity. Further analysis or sandbox testing is recommended.

The screenshot shows the 'Performance' tab in the Process Explorer properties window for process part11.exe:2372. The window displays various system resource statistics:

CPU	
Priority	8
Kernel Time	0:00:02,640
User Time	0:00:03,312
Total Time	0:00:05,953
Cycles	13,096,802,955

Virtual Memory	
Private Bytes	5,672 K
Peak Private Bytes	6,012 K
Virtual Size	178,968 K
Page Faults	4,538
Page Fault Delta	0

Physical Memory	
Memory Priority	5
Working Set	16,168 K
WS Private	0 K
WS Shareable	0 K
WS Shared	0 K
Peak Working Set	16,168 K

I/O	
I/O Priority	Normal
Reads	2
Read Delta	0
Read Bytes Delta	0
Writes	0
Write Delta	0
Write Bytes Delta	0
Other	16,531
Other Delta	84
Other Bytes Delta	1.6 KB

Handles	
Handles	1,964
Peak Handles	1,964
GDI Handles	0
USER Handles	2

## Conclusion

The file contains seven operations: TCP Receive, TCP Connect, TCP Send, Thread Create, Thread Exit, ReadFile, and TCP Reconnect. Most involve network communication (*Receive, Send, Connect, Reconnect*), while others manage threading

(*Create*, *Exit*) or file access (*ReadFile*). Network operations dominate, indicating a system focused on data transmission and connectivity.

# Advanced Static Analysis

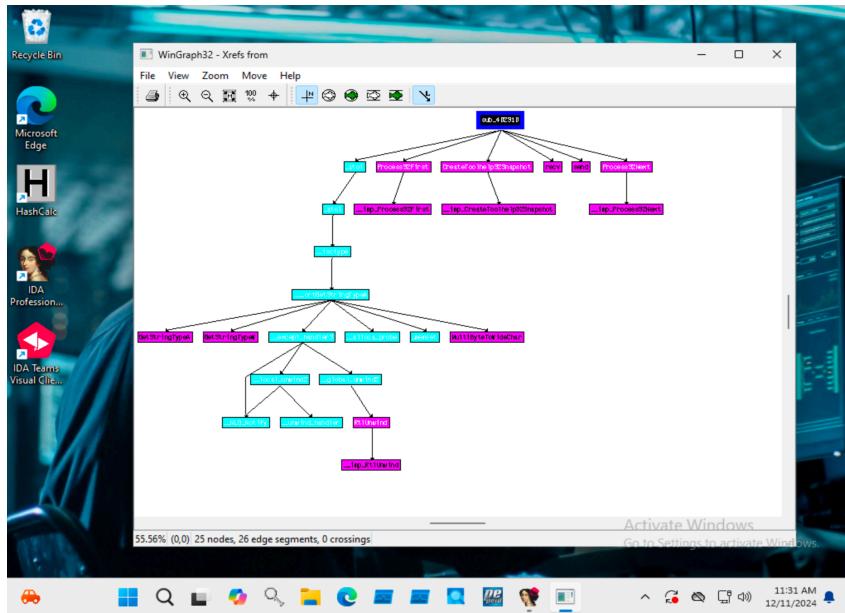
## Purpose

The purpose is to analyze malware functions, control flow analysis, and generate function and variable names.

## IDA

+ Lists processes: 0x0402310 Probably a function to provide a list of the system's active processes. It may make use of APIs like as Process32First, Process32Next, and CreateToolhelp32Snapshot.

Graph:



```
; int __cdecl sub_402310(SOCKET s)
sub_402310 proc near

String= byte ptr -236h
hSnapshot= dword ptr -234h
buf= byte ptr -230h
var_130= dword ptr -130h
var_12C= dword ptr -12Ch
pe= PROCESSENTRY32 ptr -128h
s= dword ptr 4

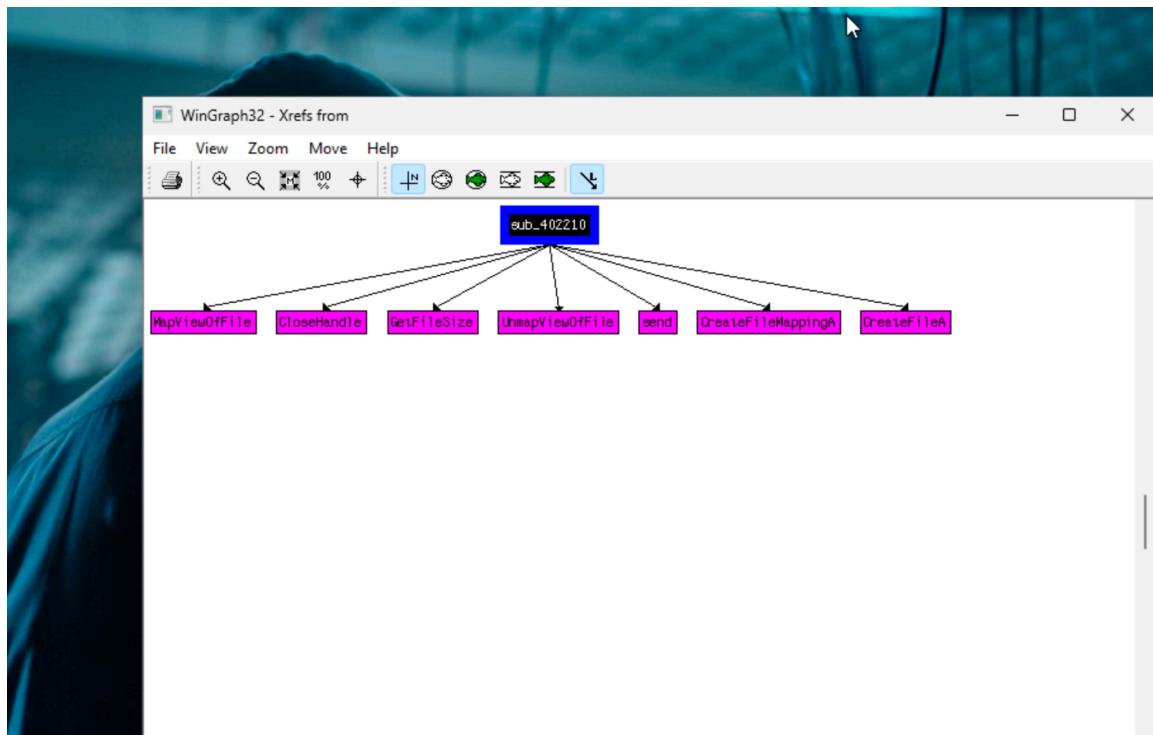
sub    esp, 238h
push   ebx
push   0          ; th32ProcessID
push   2          ; dwFlags
mov    [esp+244h+pe.dwSize], 128h
call   CreateToolhelp32Snapshot
lea    ecx, [esp+23Ch+pe]
mov    [esp+23Ch+hSnapshot], eax
push   ecx         ; lppe
push   eax         ; hSnapshot
mov    [esp+244h+var_12C], 1
call   Process32First
mov    ebx, [esp+23Ch+s]
test   eax, eax
jz    loc_4023F8
```

CreateToolhelp32Snapshot: Captures a snapshot of all processes.

Process32First: Retrieves the first process in the snapshot.

Process32Next: Iterates through the remaining processes.

+)Upload File (0x00402210): A file is XOR-encrypted using the key 0x55 and uploaded to a distant server via a socket using the sub\_402210 function (already examined).



A screenshot of a debugger window showing the assembly code for the sub\_402210 function. The code is written in Intel syntax. It starts with a prologue, initializes pointers for buf, len, s, and lpFileName, and then performs memory operations involving XOR and pushes onto the stack.

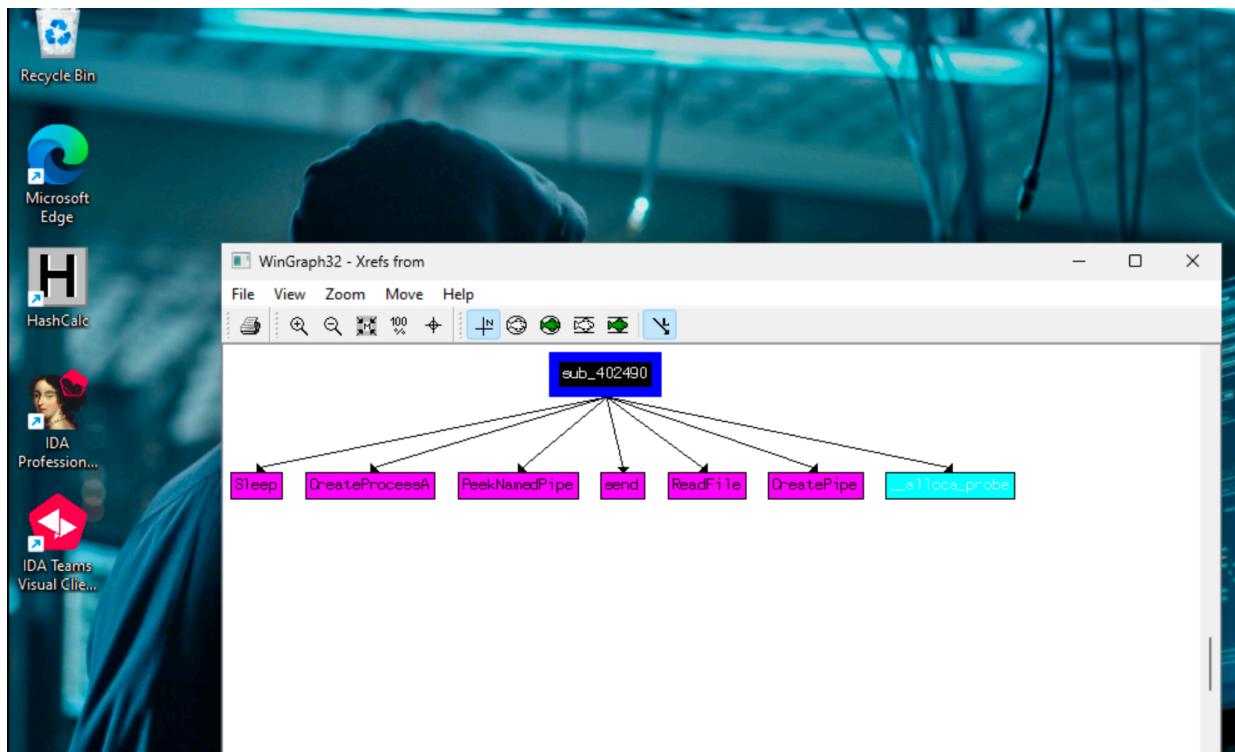
```
; int __cdecl sub_402210(SOCKET s, LPCSTR lpFileName)
sub_402210 proc near

buf= byte ptr -204h
len= dword ptr -4
s= dword ptr 4
lpFileName= dword ptr 8

mov    ecx, [esp+lpFileName]
sub    esp, 204h
xor    eax, eax
push   ebx
push   esi
push   edi
```

+ ) Remote Shell (0x0402490 and 0x0402660): By enabling a remote shell, an attacker might take control of the compromised system and run commands on it. They may transmit and receive commands via a socket or communicate with cmd.exe.

0x0402490:



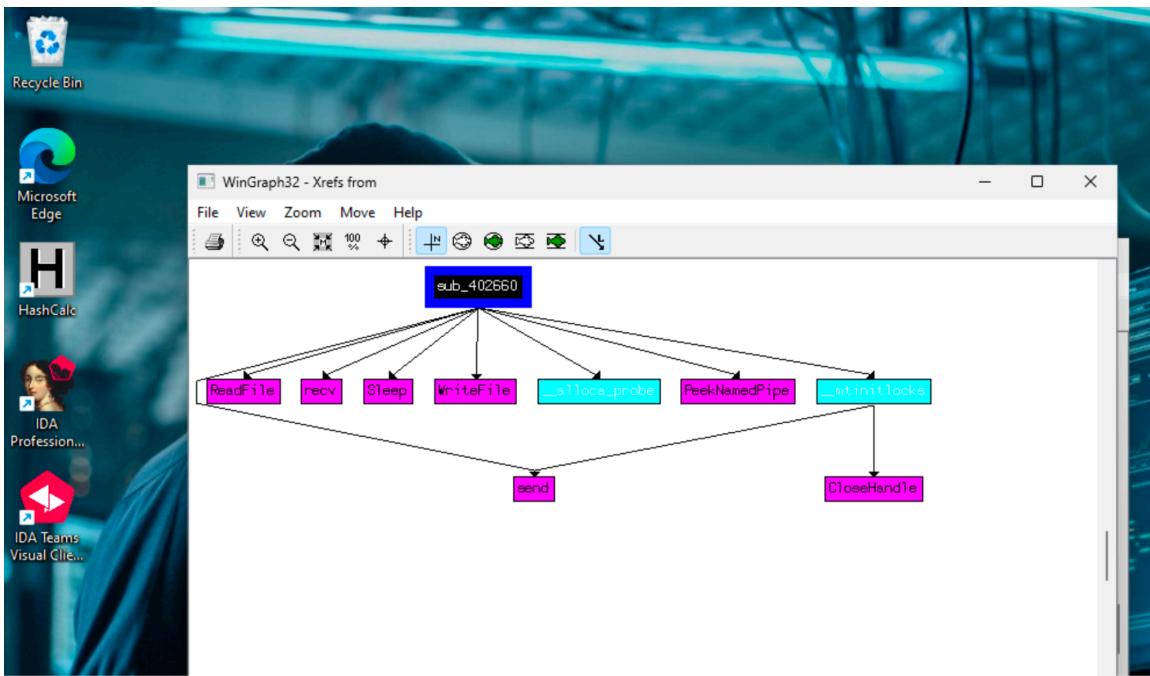
```
; int __cdecl sub_402490(SOCKET s)
sub_402490 proc near

PipeAttributes= _SECURITY_ATTRIBUTES ptr -106Ch
CommandLine= byte ptr -1060h
var_105C= dword ptr -105Ch
StartupInfo= _STARTUPINFOA ptr -1058h
ProcessInformation= _PROCESS_INFORMATION ptr -1014h
Buffer= byte ptr -1004h
var_4= dword ptr -4
s= dword ptr 4

    mov     eax, 106Ch
    call    __alloca_probe
    push    ebx
    push    ebp
    push    esi
    mov     esi, ds:CreatePipe
    xor     ebx, ebx
    push    edi
    lea     eax, [esp+107Ch+PipeAttributes]
    push    ebx          ; nSize
    push    eax          ; lpPipeAttributes
    mov     ebp, 1
    push    offset hWritePipe ; hWritePipe
    push    offset hNamedPipe ; hReadPipe
    mov     [esp+108Ch+PipeAttributes.nLength], 0Ch
    mov     [esp+108Ch+PipeAttributes.lpSecurityDescriptor], ebx
    mov     [esp+108Ch+PipeAttributes.bInheritHandle], ebp
    call    esi ; CreatePipe
    lea     ecx, [esp+107Ch+PipeAttributes]
    push    ebx          ; nSize
    push    ecx          ; lpPipeAttributes
    push    offset hFile   ; hWritePipe
    push    offset dword_40AA8C ; hReadPipe
```

Activate \ Go to Settings

0x0402660:



```
; int __cdecl sub_402660(SOCKET s, LPCVOID lpBuffer)
sub_402660 proc near

var_100C= dword ptr -100Ch
buf= byte ptr -1006h
Buffer= byte ptr -1004h
var_4= dword ptr -4
s= dword ptr 4
lpBuffer= dword ptr 8

    mov     eax, 100Ch
    call    _alloca_probe
    push    ebx
    xor    ebx, ebx
    push    esi
    mov    esi, [esp+1014h+lpBuffer]
    push    edi
    mov    [esp+1018h+var_100C], ebx
    xor    eax, eax
```

CreateProcessA: Creates a new process

WriteFile: Sends commands or responses through a pipe.

PeekNamedPipe: Reads data from a pipe to check if there's input/output.

## Conclusion

Analysis functions, such as listing active processes, enabling remote shell access for control, and uploading encrypted files to a remote server.