# Windows malware reverse engineering using Ghidra

## Executive Summary

With the rapid development of network, the networking program is becoming popular, so as the malware. Hackers are trying writing malware that can not be cracked. They want to have their malware to be safe, so the interpreter language is not the first choice to writing malware. Static linked programming language is more prefered. Golang is a static linked programming language that can build a very huge static binary that can be easily transform to another platform, such as Windows, linux, android and even macos. Static linked binayr is vary large and it’s hard to do reversing engineering. So this language is usually the first choice for the hacker to write malware.

## Introductions

In this document I will give a brief reversing engineering of the golang binary malware by using ghidra decompiler.

## Environment

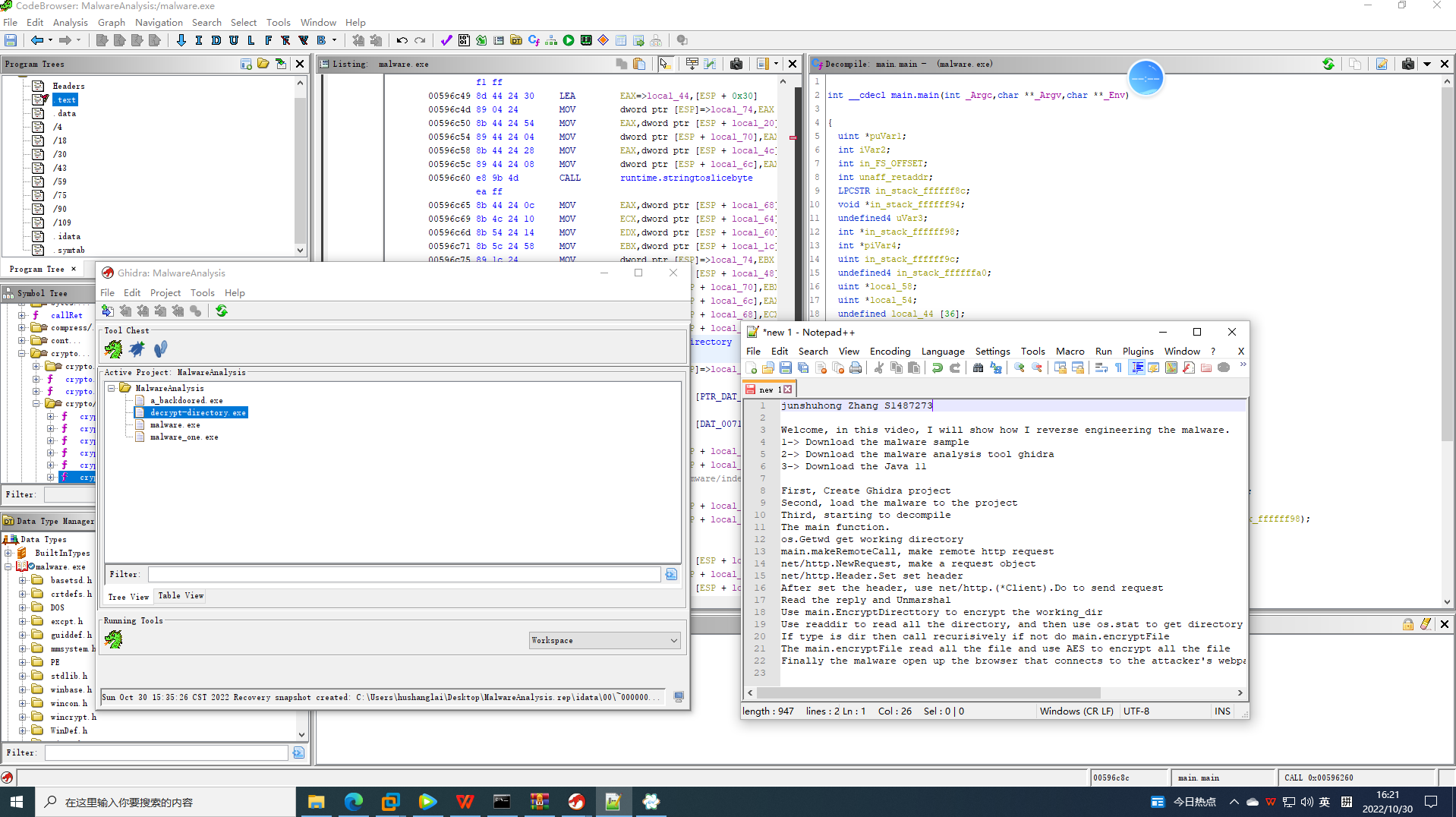
System: windows 10 home edition

Decompiler: Ghidra 10.0.1

Editor: Notepad++ 7.8.2

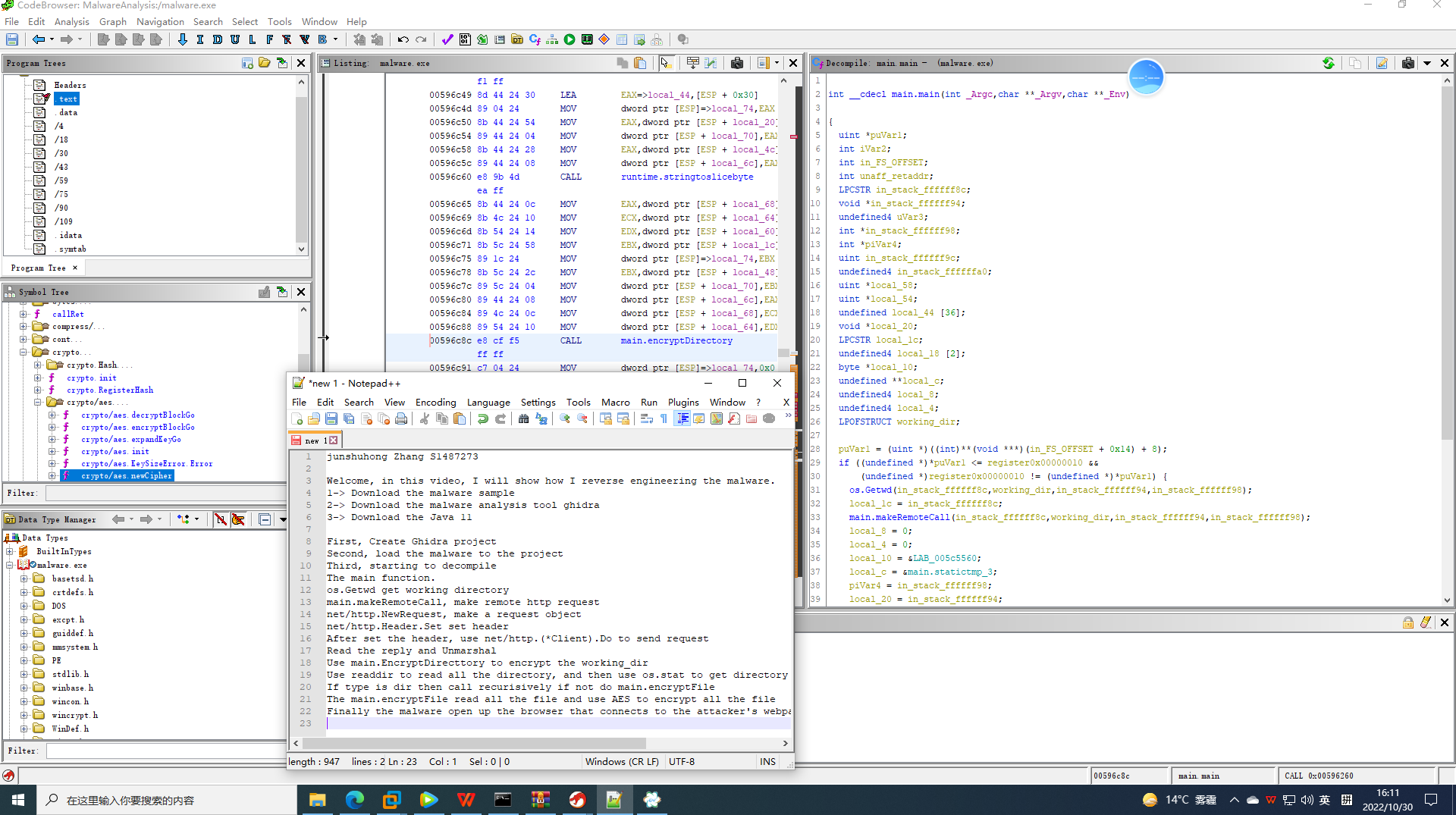
## Ghidra project

Create a project that named malware analysis

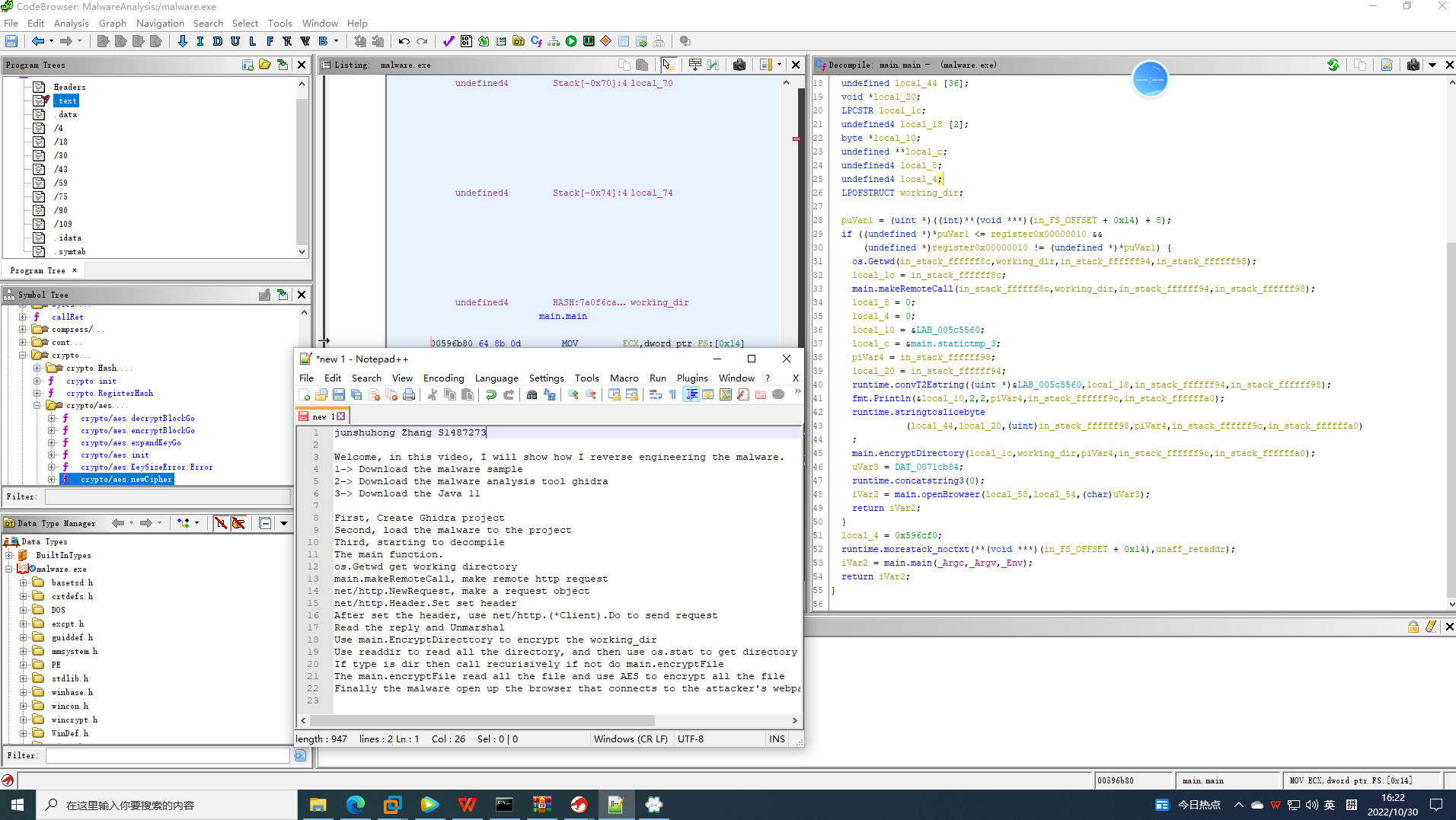


## Find main function

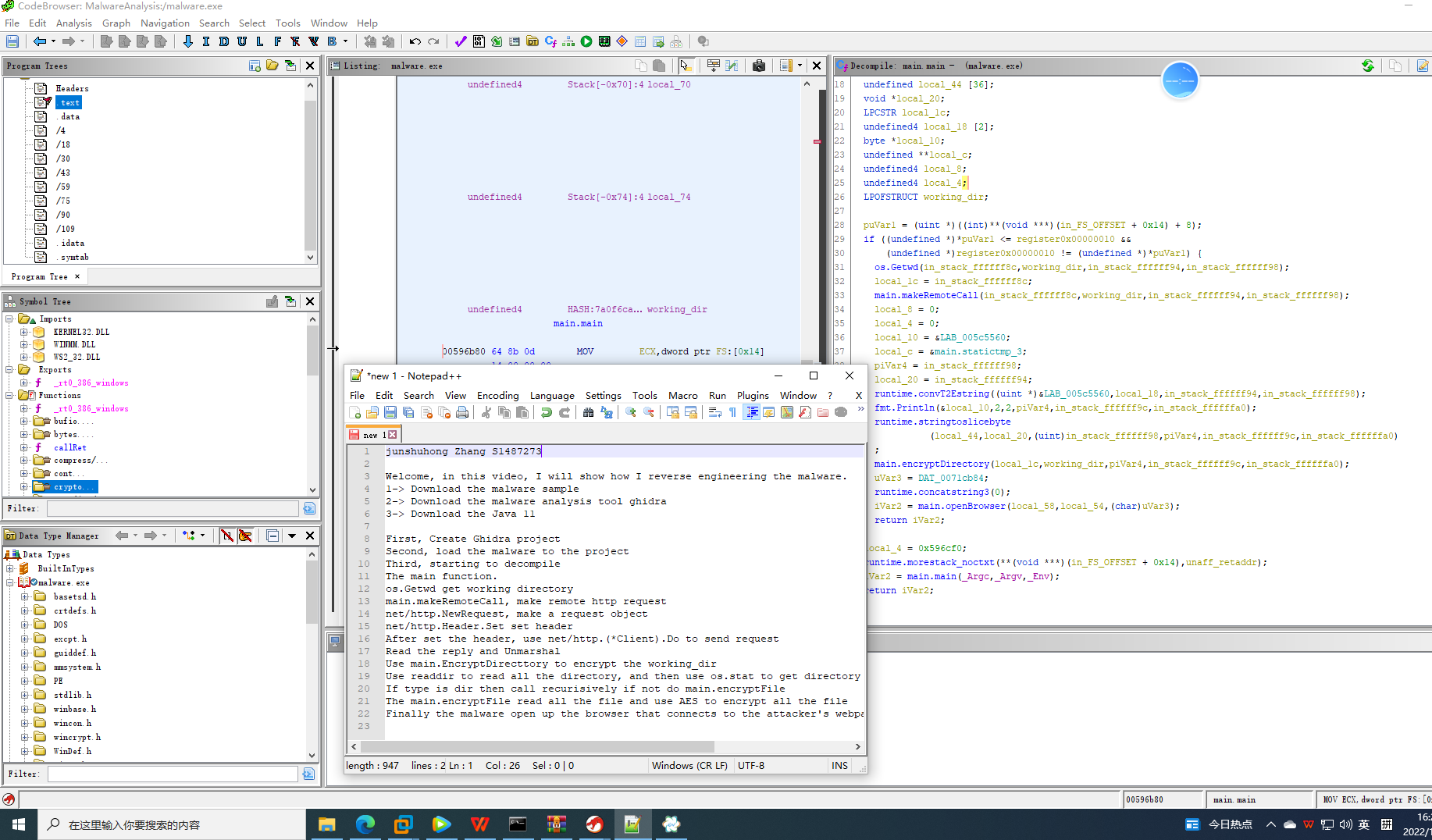
Main function is all the entry of the program, program start to execute code from the main function.



from the main function we can clearly see what the program do.

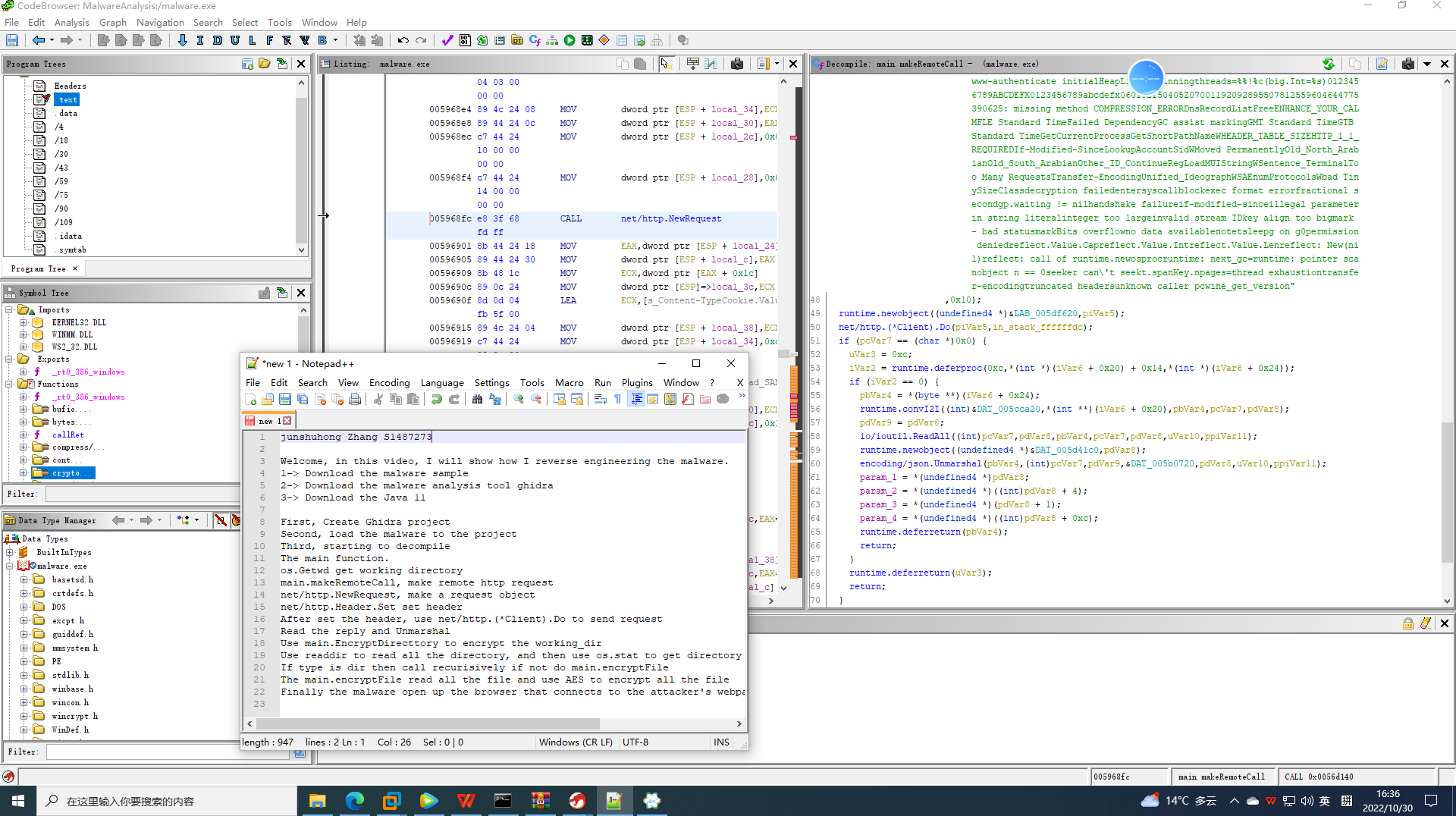


A

The left side is the symbol tree, which contains imported symbols and exported symbols. 

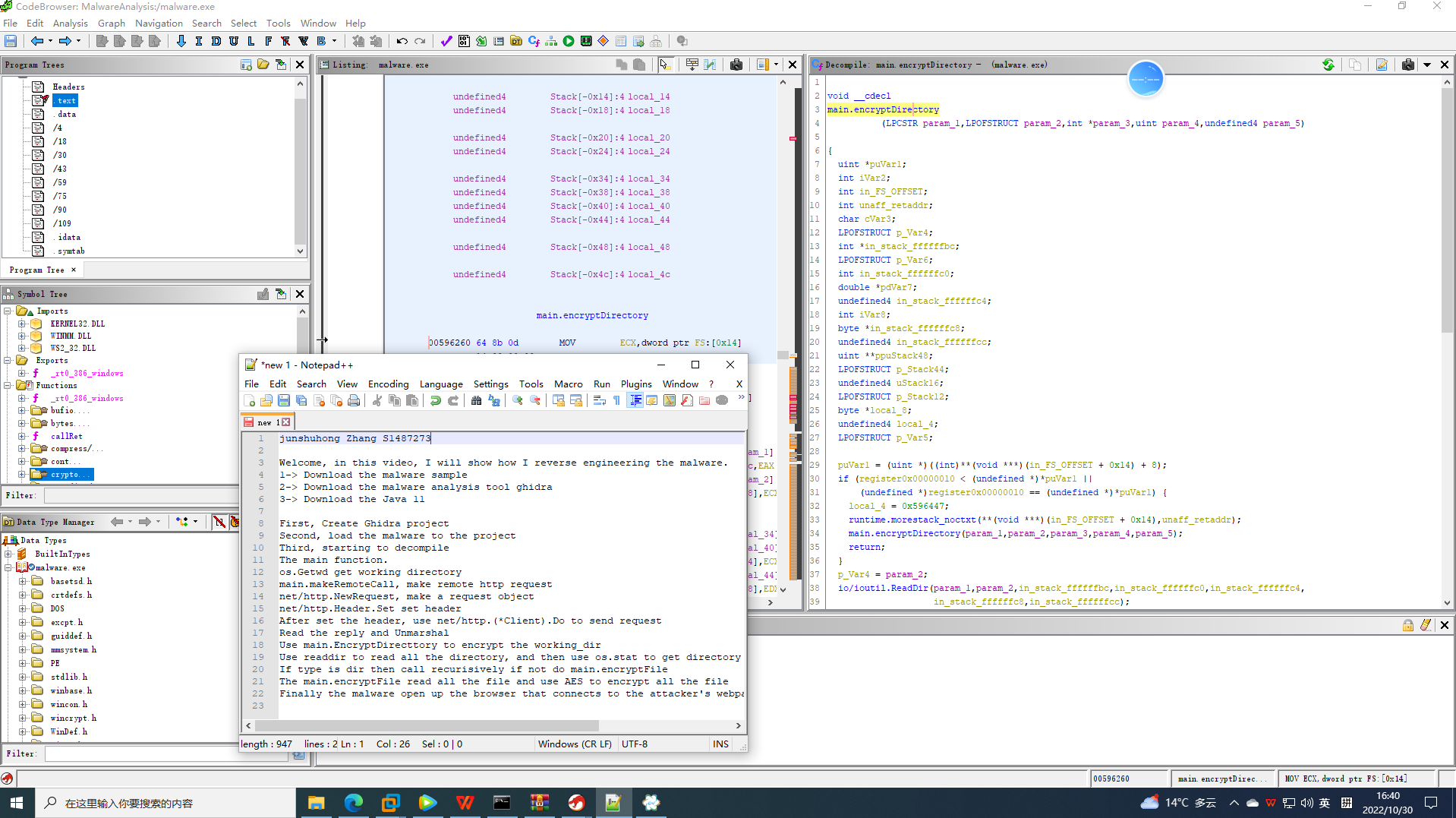
## Create HTTP Payload

Use the function that provided by net packet to construct the HTTP payload. After construction, use Client.Do to send the HTTP request. Finally use unmarshal function to store the reply.

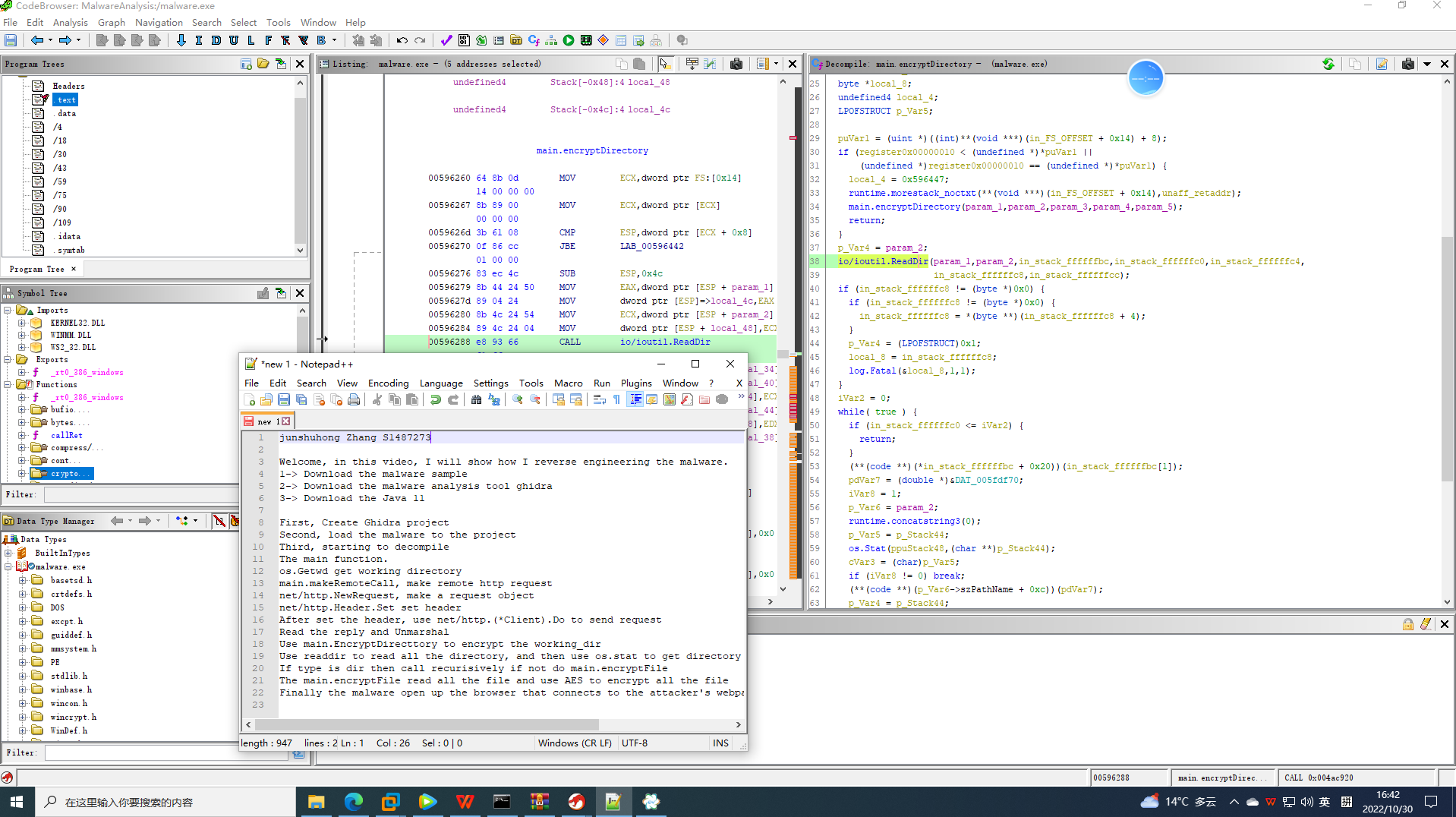


## Reverse the encryptDirectory function

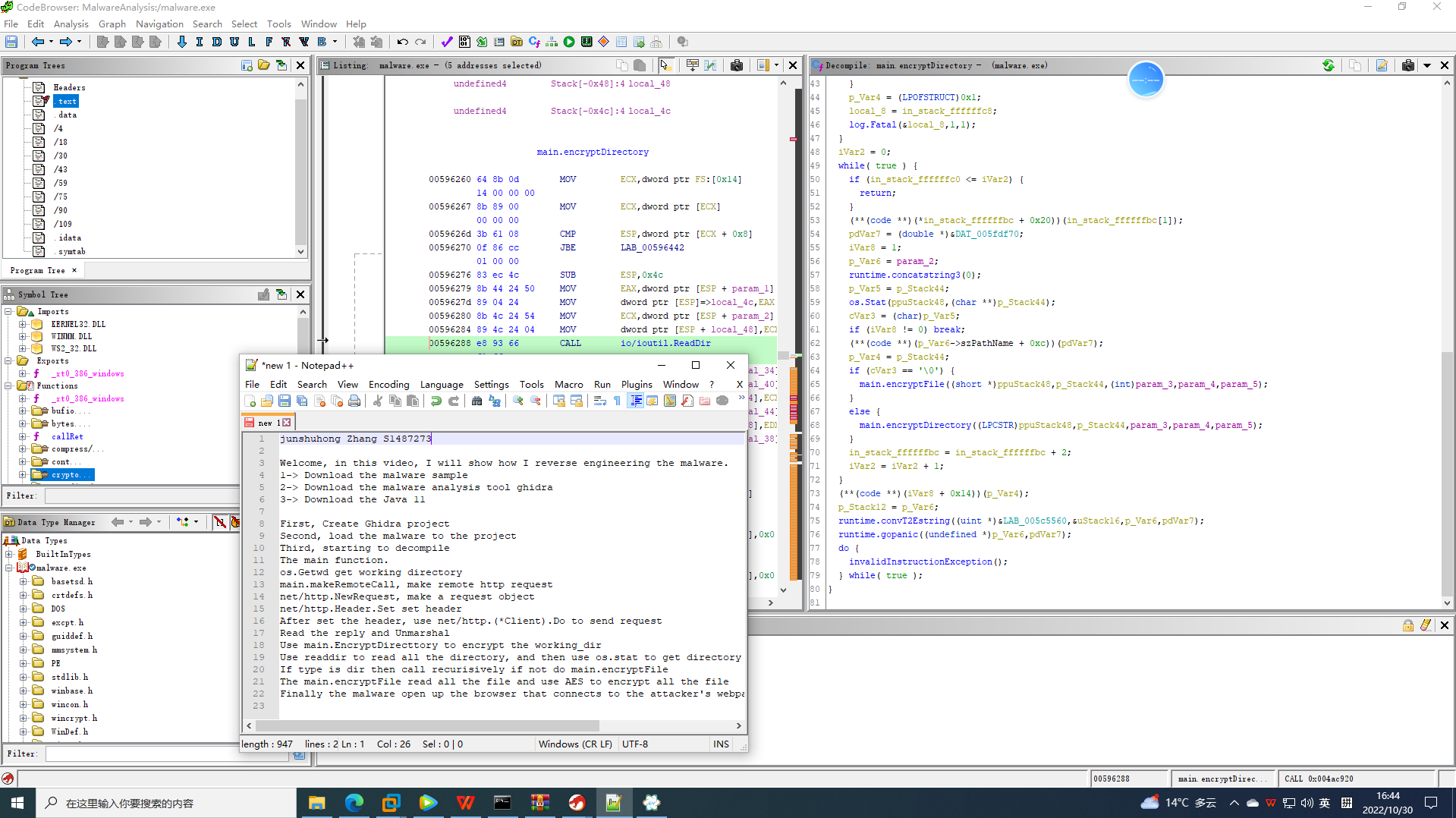
The encryptDirectory function is simple. As its name show that it’s used to encrypt the directory.



This function use readDir function provided by io/util packet, which is used for reading directory.

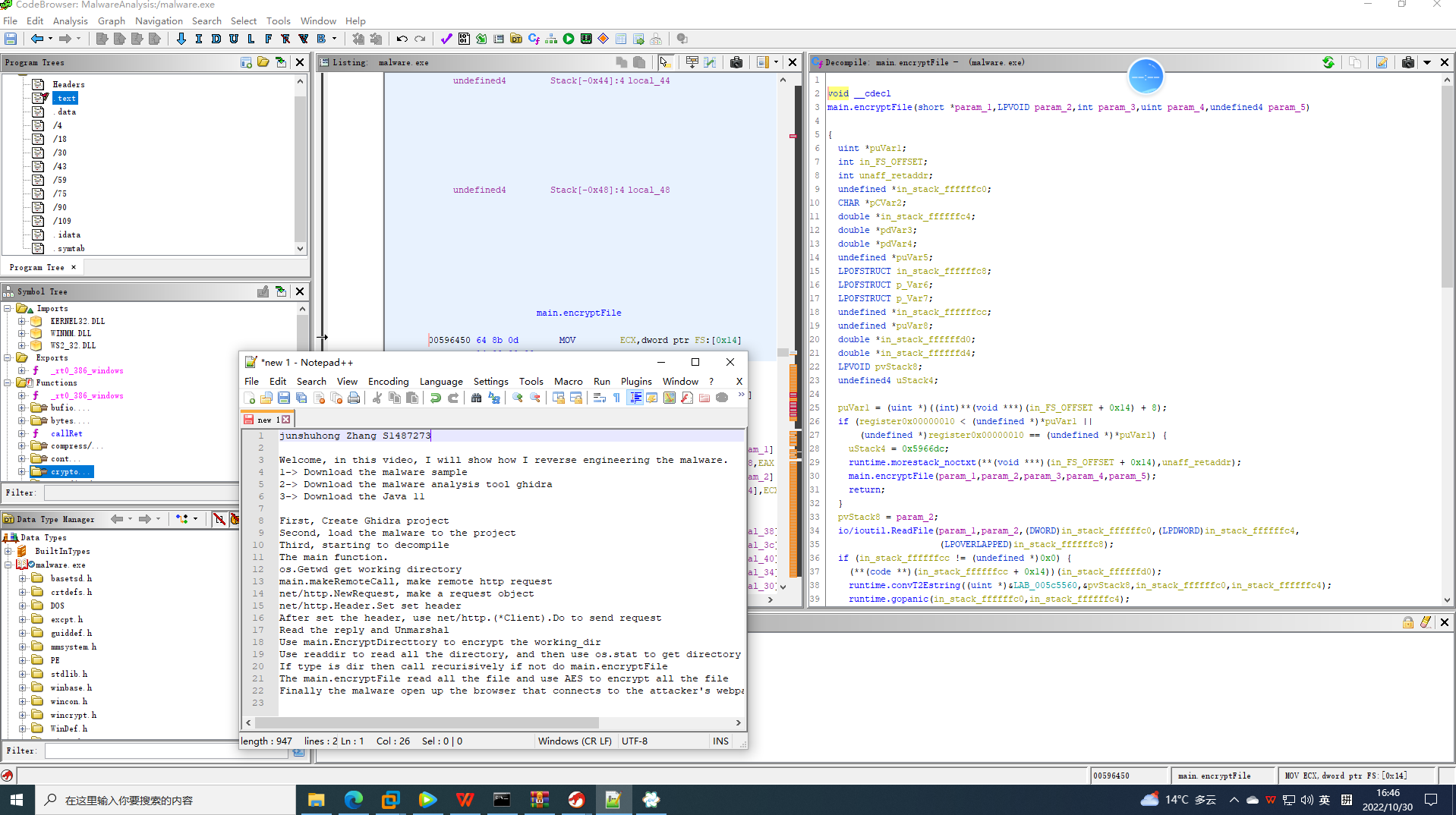


Then judge the file type, do different operation on those two different file type. Do encryptDirectory for the directory and do encryptFile for file.

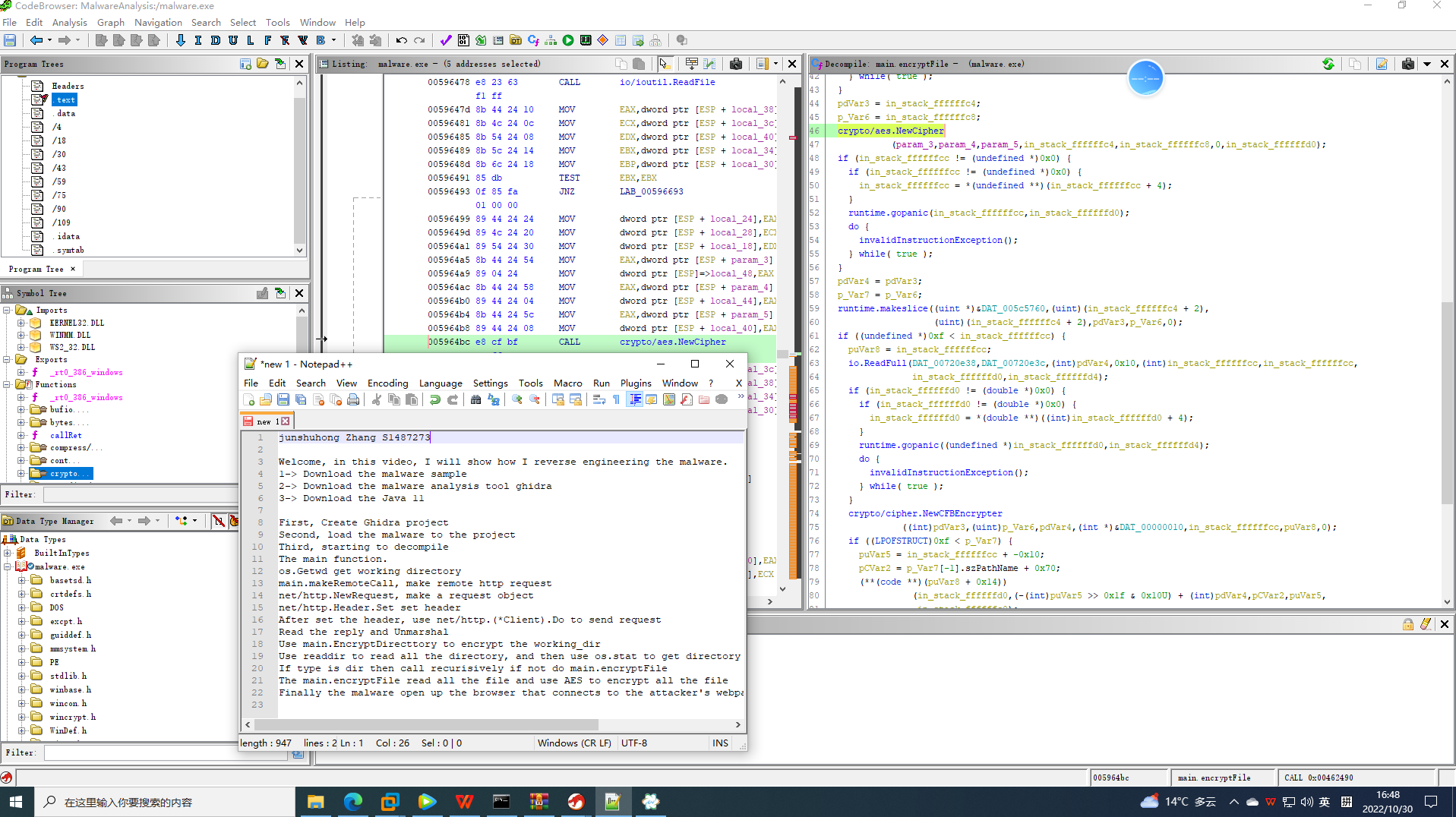


## EncryptFile function decompilation

First, this function perform ReadFile function to read file.



After open up the file. This function new a AES cipher and the new a AES encryptor to encrypt the content of the file and filnally use the encrypted data to replace the normal data.



## Conclusion

First we download the malware analysis tool Ghidra 10 and it’s dependencies Java 11.

Second we create a project for the malware analysis for future analysis easily.

Third, analyze the malware and see what it do.

1. The malware connect to the remote hacker’s host
2. Iterate all the file and call encryptDir function.
3. For every items call ecrytpFile if it’s a file or call encryptDir recursively.
4. Call openBrowser function to open web browser and jump to the hacker’s page.