IDA Pro Disassembling of Lab01-02.exe

1. Start Ida pro and select lab01-02.exe
   1. A diagram of a diagram

      Description automatically generatedA screenshot of a computer

      Description automatically generatedLook at graph view to see structure of functions and how their connected:
   2. Looking at the function circled you can see where the Packing executable ends and where the real malware package begins:
      1. Add a break point onto the jmp near ptr dword\_401190 for manual unpacking in the debugger.

A screenshot of a computer program

Description automatically generated

* 1. After looking at the graph it is beneficial to also check the imports section to see what the malware is attempting to do.

A screenshot of a computer

Description automatically generated

* + 1. Looking at the imports you can see that lab01-02.exe is packed(above)
    2. In this import section you can see LoadLibraryA, GetProcAddress, createServiceA, and InternetOpenA which is more evidence that the malware is packed
       1. These imply that the malware is going to open the internet and create a service

1. Using text view:
   1. You can see that the lab01-02.exe was packed with UPX:

A screen shot of a computer code

Description automatically generated

1. View after unpacking this malware with UPX (There are ways to manually unpack this malware but since we know it is using UPX we can unpack it using that packer. To unpack manually use tools like PEditor and Scylla to dump the real executable from ram into a file here is a good video on how to do so: <https://www.youtube.com/watch?v=vvk_ISkKOAE>):
   1. View of functions after unpacked:

A screenshot of a computer

Description automatically generated

1. Looking at new uncovered functions(below):

A computer screen with text on it

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

This is the sub\_401040 function

1. A screenshot of a computer program

   Description automatically generatedA screenshot of a computer program

   Description automatically generatedA screenshot of a computer program

   Description automatically generatedWithin these functions it is important to look at the CALL and JMP commands as these are used to go to or call another function or process which is used by the malware.

This is the startAdress function

This is the main function

This is the start function

* 1. Looking at the main function you can see that it uses the CALL command to call sub\_401040 which is the where the startAddress function is. Within the start address function you can see it calling the imported functions InternetOpenA and InternetOpenUrlA. The Actual URL that is being opened is within the szUrl variable(this variable is see within the startAddress function too): [www.malwareanalysisbook.com](http://www.malwareanalysisbook.com)
  2. For even more specifics you are able to use IDA Pro to decompile the assembly into a higher level language. This functionality can be very helpful for determining what a malicious sample is doing, keep in mind that this extrapolation upwards is only a guess as IDA pro is using estimation to generate these Screenshots that are bellow:
     1. To get the sudo code to show is by pressing f5 in the function.

A computer screen shot of a program

Description automatically generatedA computer screen shot of a program

Description automatically generated

A screen shot of a computer

Description automatically generated

A computer screen shot of a program

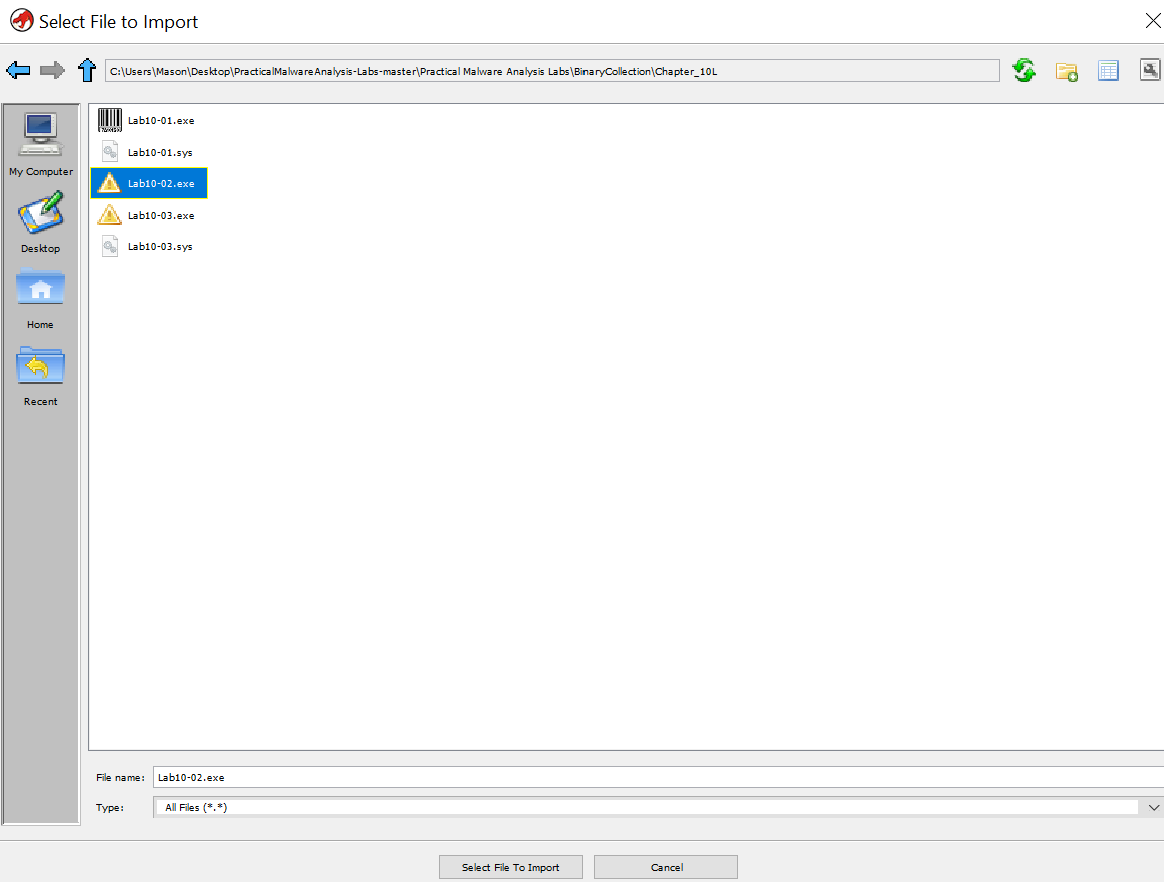
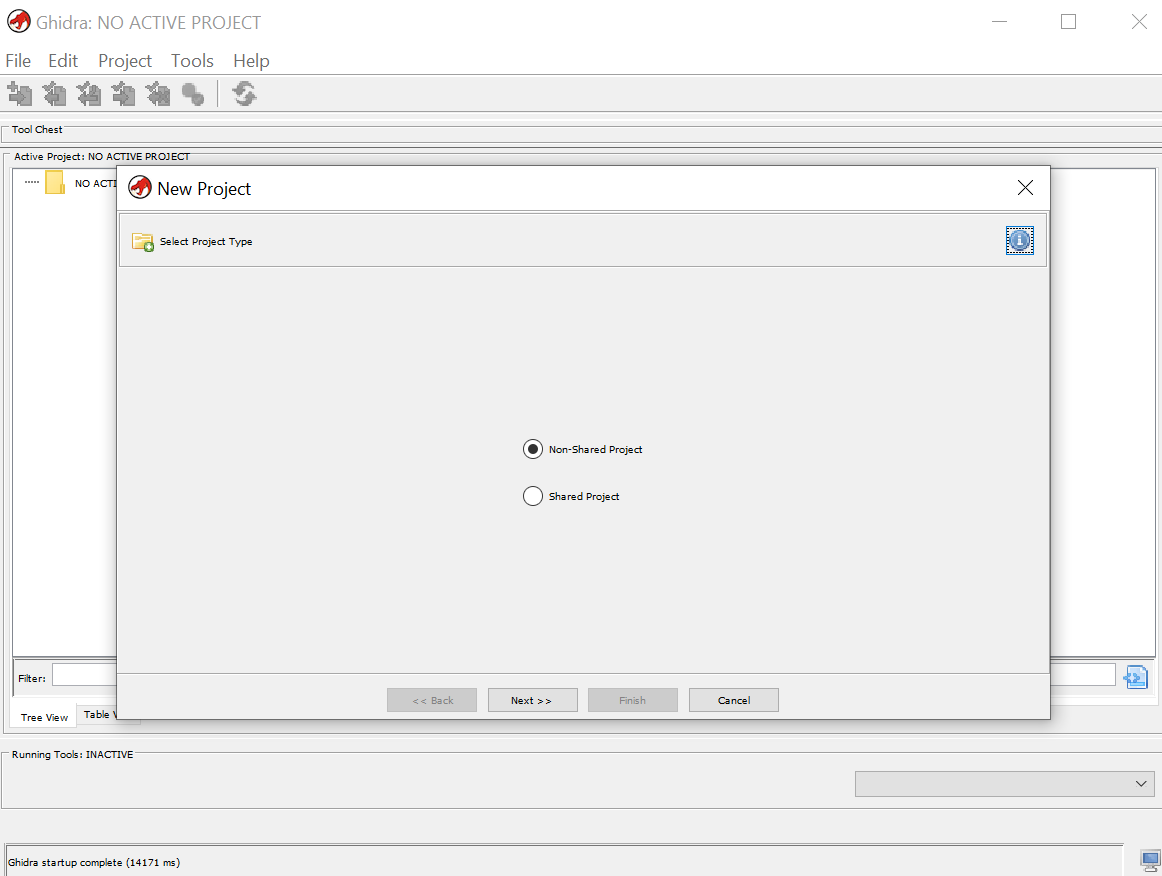
Description automatically generated

1. A screenshot of a computer program

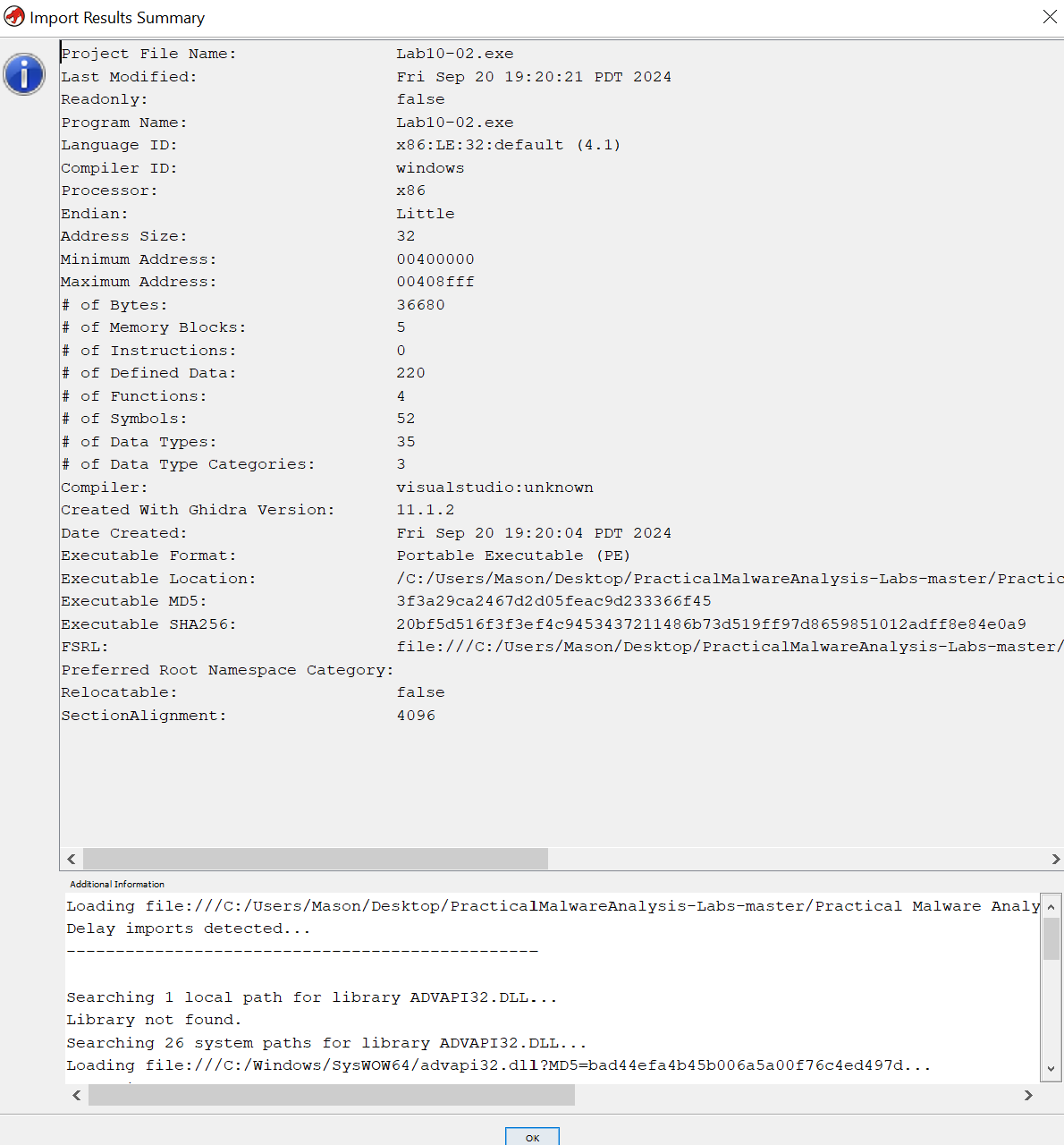
   Description automatically generatedhow debugger works in detail09
   1. The IDA pro debugger allows for breakpoints to be set and for the user to step through the code to see what the malware would do if it was run on an unsuspecting machine.
   2. The debugger has a few key windows that allow for better and easier analysis, the module section shows all DLL imports that have been accessed by the .exe
   3. the threads section shows all threads generated by the malware and allowing you to double click into them allowing for further investigation
   4. The register view shows all values and variables currently in them helping with tracing specific commands or variables
   5. The stack view displays local and temporary variables, incoming arguments that are used in calling, saved volatile registers, and addresses

Ghidra Disassembling of Lab01-02.exe

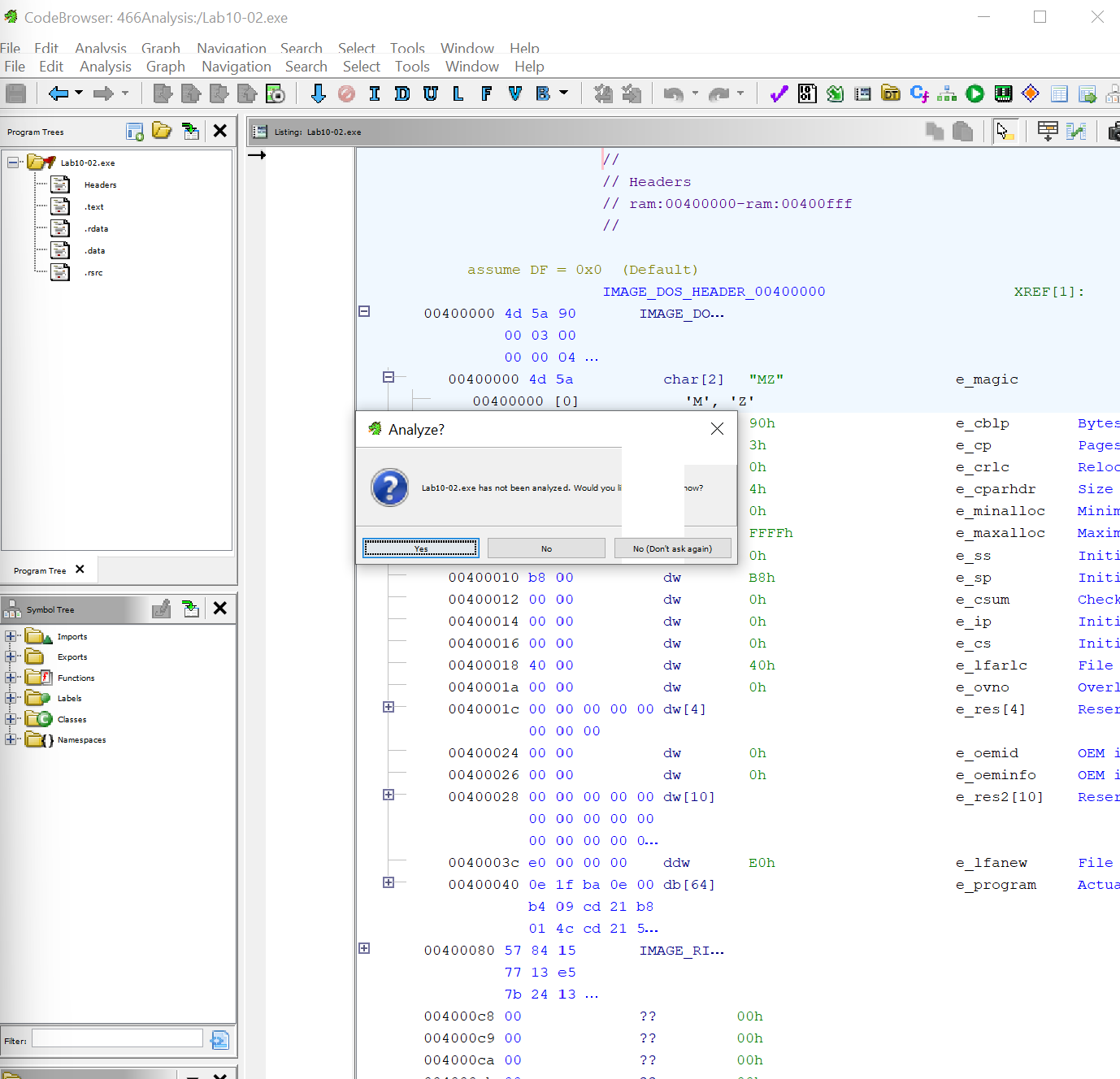
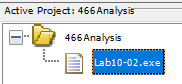
1. Open Ghidra and create a new project, then import the selected file



Import Results

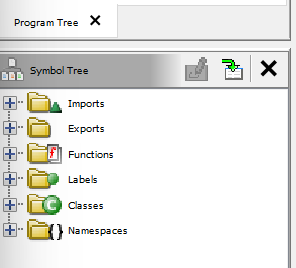
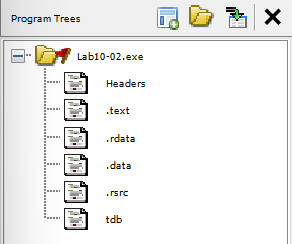


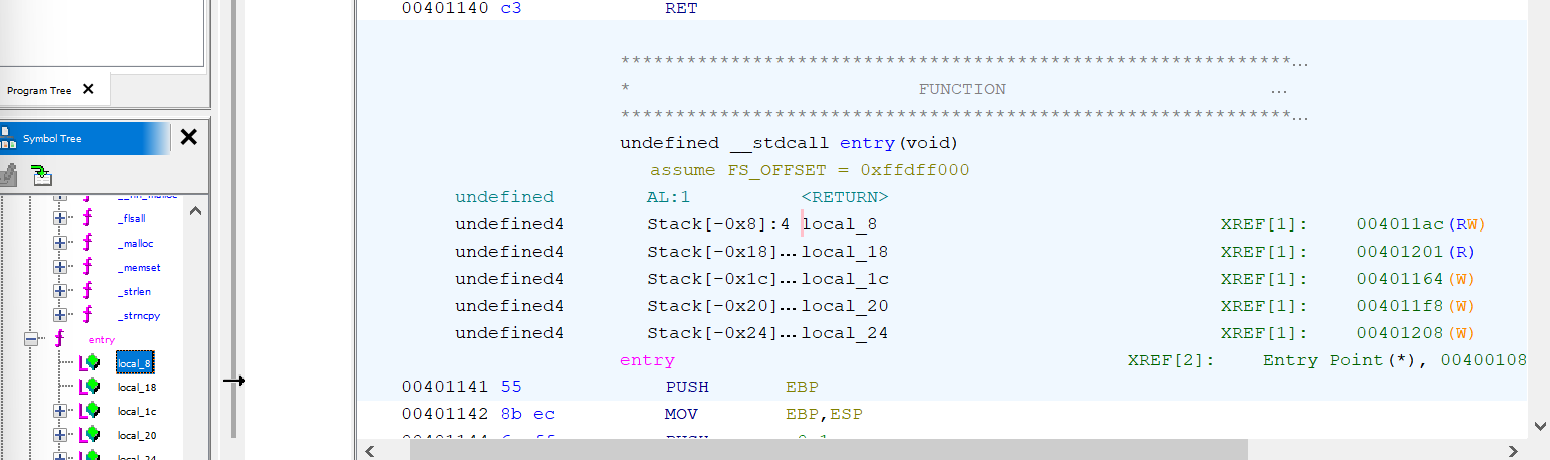
1. Double click the malware file to analyze, the code browser will open, click yes

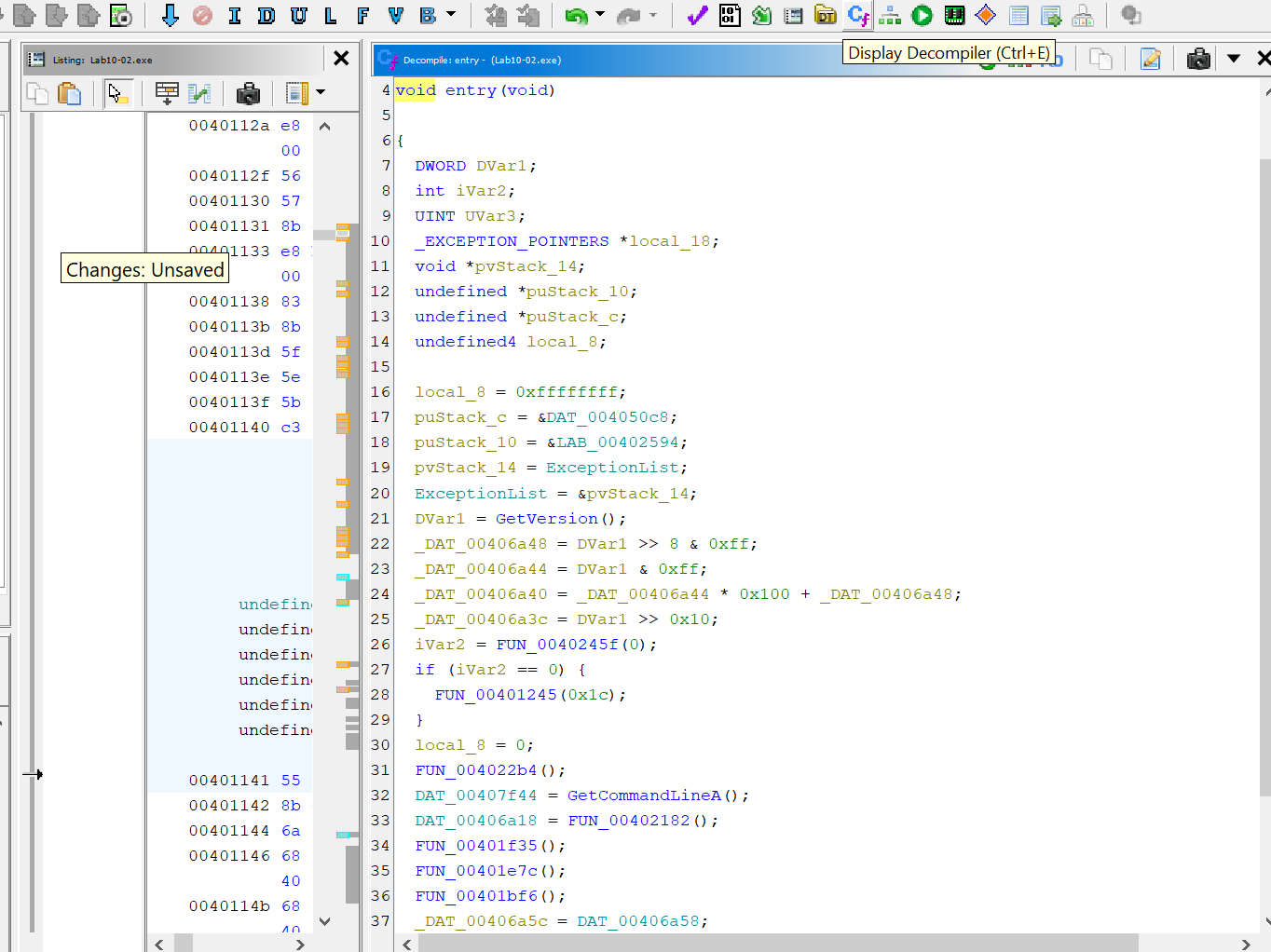


1. Analysis Options will open, select “WindowsPE x86 Propagate External Paramaters” to help analyze function parameters, click apply then analyze

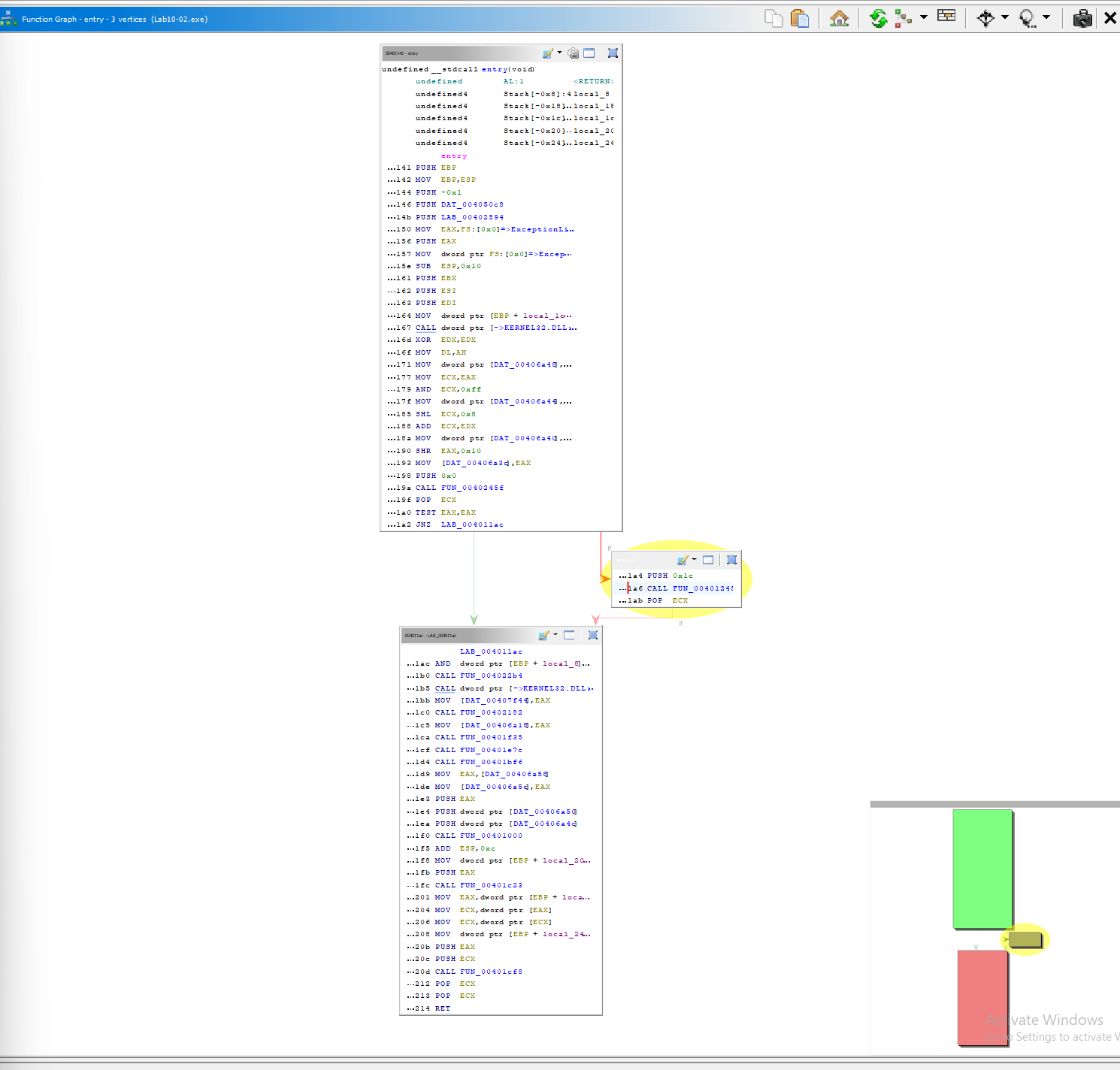
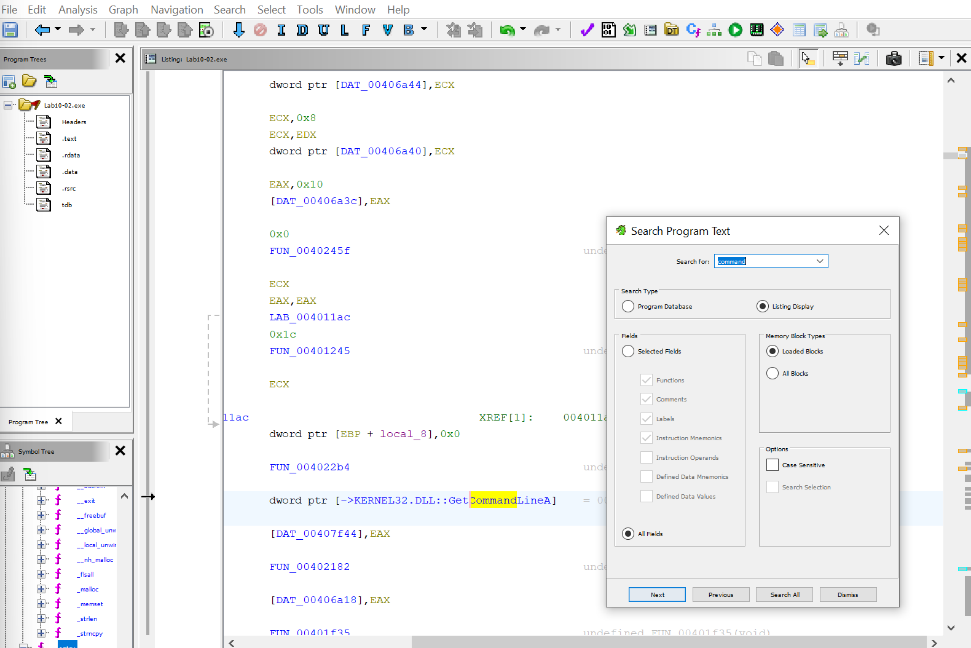
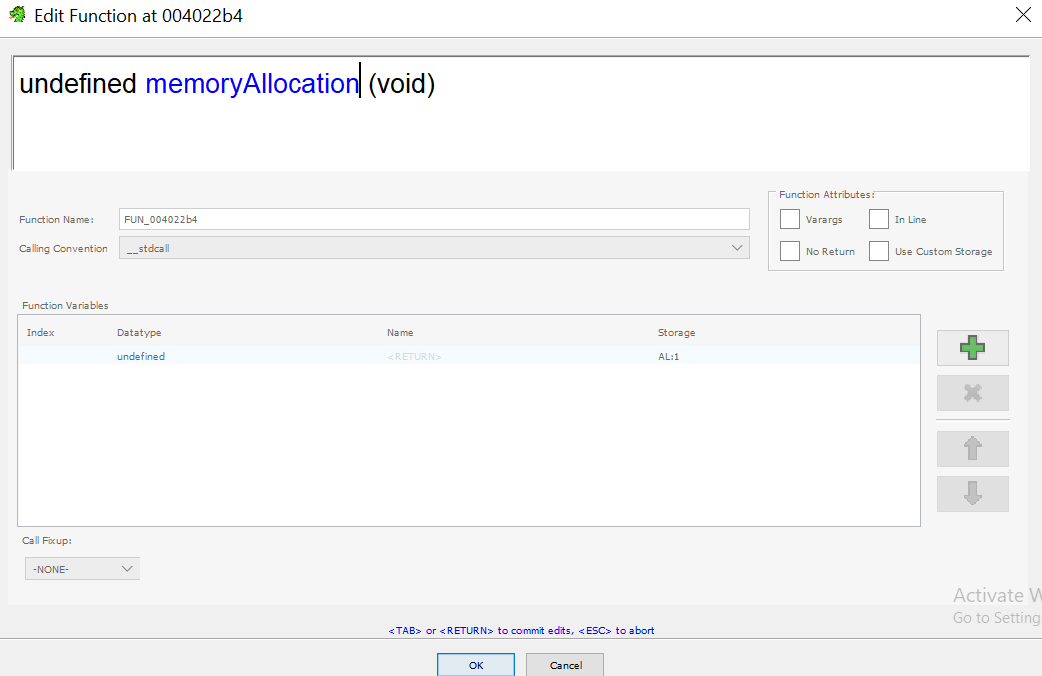
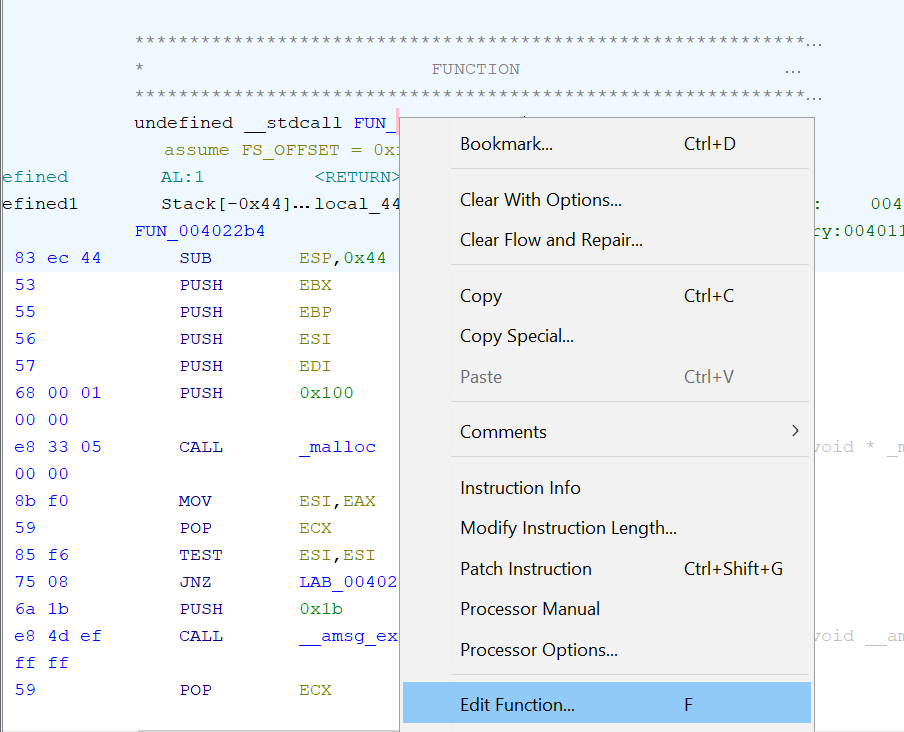
Program Trees and Symbol Trees help parse out the file's headers, imports, exports, functions, etc.



1. By going to “entry” in the Symbol Tree, you can see the entry point of the malware
2. Click “Display Decompiler” to show Ghidra’s attempt at decompiling the selected code into C



You can see the entry function is an initializer by its conduction of base tasks such as getCommandLine and getVersion, while calling many other functions

1. Click “Display function graph” to show a visual representation of the flow of the malware 
2. You can search by clicking “Search” in the toolbar and clicking “Search Program Text”. This will search through the program and find matches for what you type. In this instance I searched for any references to “command” to see how the malware is accessingthe command lline. 
3. Change function names to log functionality. Right click, Edit Function, save changes 

x64dbg Disassembling of Lab01-02.exe

1. x64dbg has a 32-bit version and a 64-bit version depending on what type of bit the malware is. In this case Lab01-02.exe is 32-bit so open it with the 32-bit version.
2. When you hover above this icon you can see the prompt to run user code. This will take you to the address of the entry point.A screenshot of a computer

   Description automatically generated
3. After this you can step through the program to see how it works while creating breakpoints for significant portions.

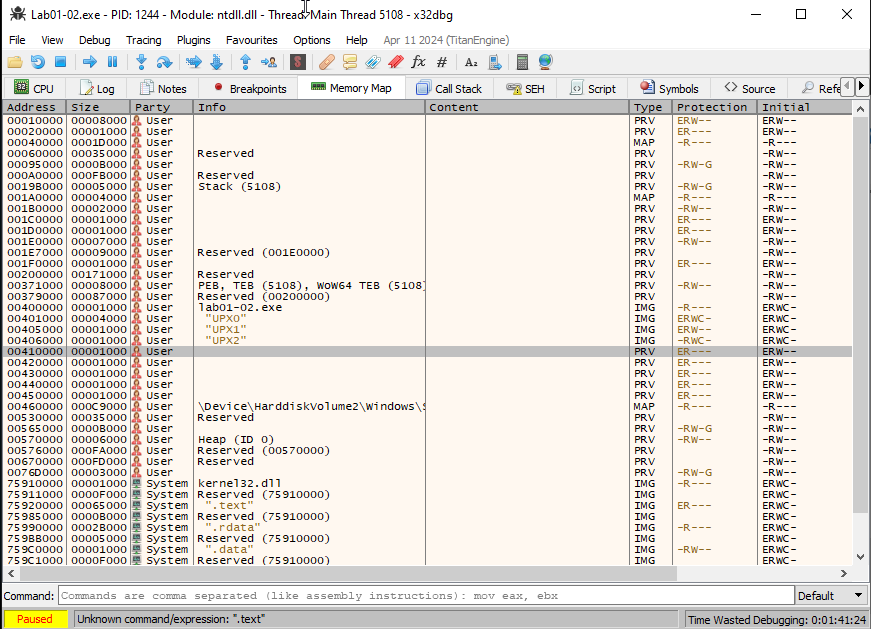
A screenshot of a computer

Description automatically generated

1. This file is packed but you can execute the program within x64dbg to unpack it.
2. If you right click on the cpu tab contents you can search for the intermodular calls and investigate all the calls made.

A screenshot of a computer

Description automatically generated

1. 

Here in memory map you can see the exe has UPX, so it is packed.