CS4379/CS5375
Dr. Jaime C. Acosta
Software Reverse Engineering

Homework Assignment – Dynamic Analysis

Due Date: Before 11:59pm on **September 29, 2022**

This assignment will be done in groups of 2-3. Late work will only be accepted until October 1st at 11:59pm and will be deducted two letter grades.

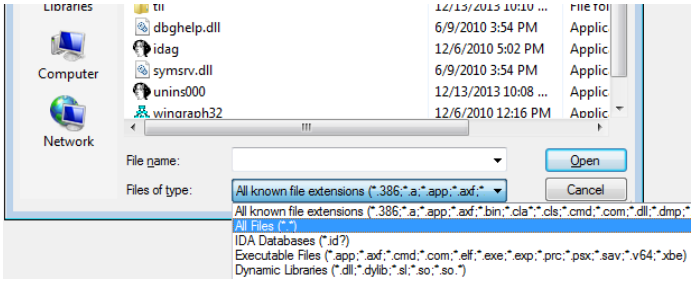
When answering each question, be very explicit describing your steps and include any screenshots or other materials you think will help evaluate your work.
Use your environment VM to complete this assignment. All files required for this assignment can be downloaded from the course web page:
https://cs5375.cs.utep.edu/

Part 1: Obtain the binary

1. Download/extract the file called Trigger.zip, using 7zip, from the course web page. If your browser removes the file, try a different browser (chrome seems to work fine). Use the password: *infected* to decompress. This will extract two files (trigger.exe and cygwin1.dll). The dll file must be in the same directory as trigger.exe at all times or else it will not execute.

Part 2: Ensure you installed IDA Pro 5 as Administrator (otherwise debugging won't work)

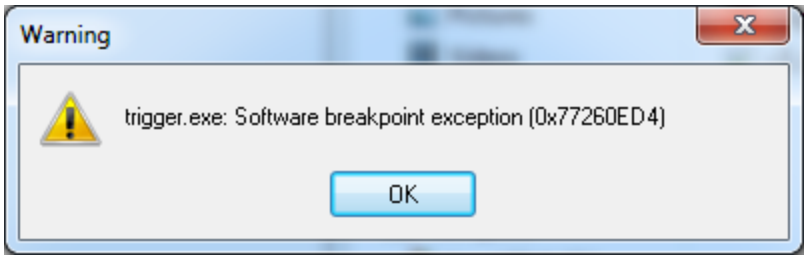
1. Start IDA Pro 5 as ***Administrator***
2. Using the interface, select the analyze the trigger.exe binary that you decompressed in **Part 1**.
3. Open IDA Pro by right clicking on the icon and selecting ***"Run as administrator"***
4. Click on Go
5. Click on File->Open
6. Navigate to the directory where you extracted the zip file.
7. Select to view all file types and choose the trigger.exe file.



8. Disassemble!

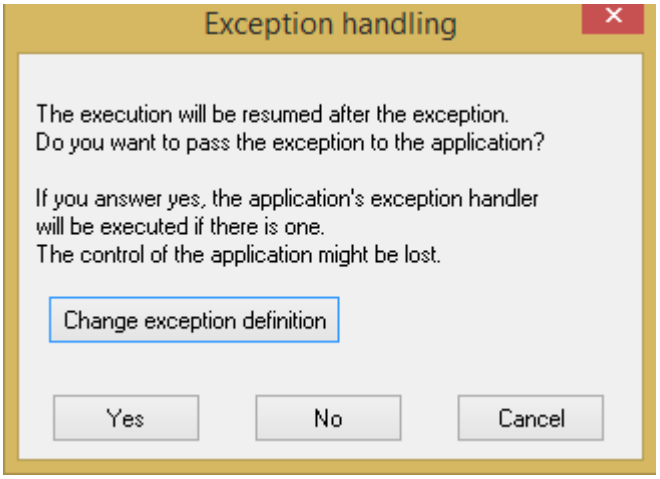
Part 3: Debugging with IDA Pro 5

1. When you start debugging you will receive a prompt like the following:

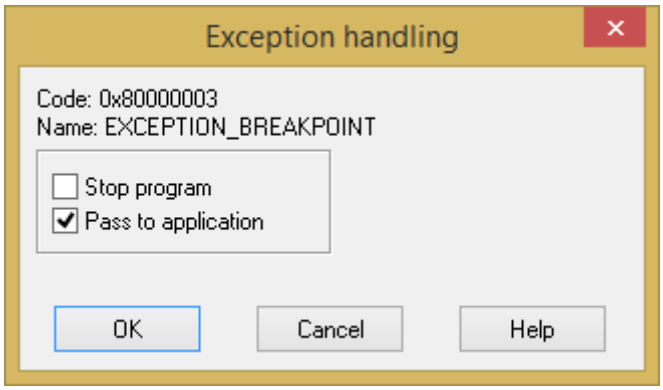


2. Press **OK** and then continue the debugging process (press F9 or press the play button).

Next, you will receive the following prompt:



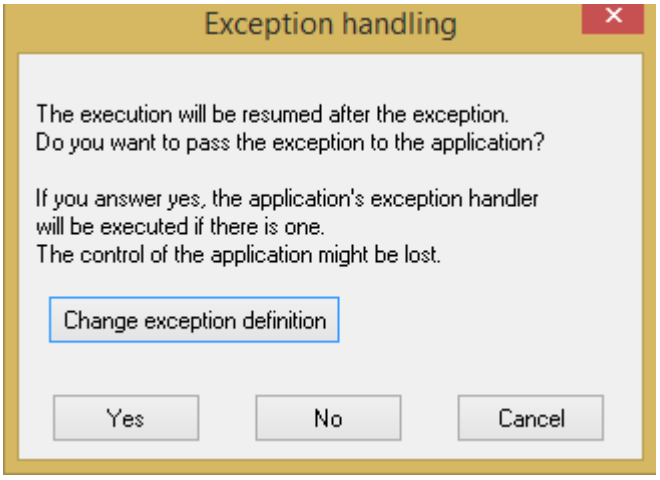
3. Click **Change exception definition** and then make the following selections:



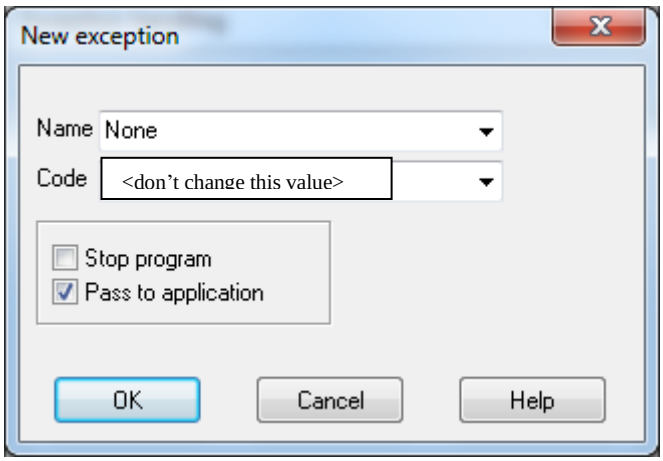
4. Press **OK** and then **Yes.**

5. Continue the debugging process (press F9 or press the play button).

If, at this point, you get the following window **again**:



6. Click **Change exception definition** and then set the **Name** to any text (the example below shows it changed to "None"), then **uncheck** "Stop program" and **check** "Pass to application" (don't change the address in the Code field):



7. Press **OK** and then **Yes.**

8. Continue the debugging process (press F9 or press the play button).

Repeat this process (using a different **Name** – any name will work) if you encounter additional exceptions as your debug.

Part 4: Assignment

Answer the following questions:

1. What is the 1st secret passphrase?

2. What is the 2nd secret passphrase?

3. What is the 3rd secret passphrase?

4. What is the 4th secret passphrase?

5. What is the 5th secret passphrase?

Complete the following:

6. Write java code to implement the 1st passphrase check in the same way as the assembly code.

7. Write java code to implement the 2nd passphrase check in the same way as the assembly code.

8. Write java code to implement the 3rd passphrase check in the same way as the assembly code.

9. Write java code to implement the 4th passphrase check in the same way as the assembly code.

10. Write java code to implement the 5th passphrase check in the same way as the assembly code.

Deliverables: Email a zip file with the subject **Assignment-DynamicAnalysis** to fall22cs5375@gmail.com The zip file must contain the following:

- For questions 1-5, a Microsoft Word document with written steps and screenshots detailing your thoughts and actions during your completion of each question.
- For questions 1-5, your idb file that contains your comments, renamed functions, etc.
- For question 6-10, a documented java file called Passphrase<#>.java, where <#> is the passphrase number.

  Important Note: Your java code should make the checks for the solutions in the same way as the assembly; to the ***extent that it is possible with Java***. For example, do not simply implement a string compare, unless that is what the binary does. If there are multiple solutions, your code should also allow for multiple solutions. Lastly, your grade will factor in the readability of your code; use understandable variable names, good comments, and good coding style.

Note: No credit will be given if you simply supply an answer, you must document your steps and your code in sufficient detail.