Assignment 1: Dissecting Malware

Context:

You are an administrator of a system. After you find out your system is compromised, you find a malware binary left in the system. As you were curious about the malware, you run the program. As usual, the malware just terminates without doing anything.

Can you figure out how to run the malware, so that it would do malicious activities?

Description:

You are given a fake malicious program (the sample program hereafter). It does a suspicious task, but it does not harm your system. However, it is not recommended running this program on machines managed by the university.

- However, the activity of this sample program might be considered as suspicious/malicious by some programs or administrators. If you are running it on one of the university-owned machines, and if you are contacted by anyone regarding this sample program, you should tell me asap.

Given the sample program, you need to create your own pintool to analyze the program. Specifically, you need to identify what are the suspicious actions, what are the specific conditions the sample expects. You are allowed to use the following tools.

- Debuggers such as gdb
- Disassemblers such as IDA, Ghidra
- Any pintool available on the Internet

Here are some specifications for the program. *Note that in practice, you will never get such information and it is up to you to figure out those.*

- The binary is statically compiled by gcc. This means that it would not make a library call.
- The sample program is packed by the UPX packer (https://upx.github.io/). If you do not know what is a "packer", please read https://en.wikipedia.org/wiki/UPX.
- Due to the packer, symbols are not accessible. You need to look at the instructions to figure out the semantics of the program.

What to do:

1. Download the Pin and install (i.e., compile) it – check the lecture slides for this.
2. Download the sample file.
3. Make your own pintool to analyze the sample.
4. Write the report as described below.

Do not do:

1. UPX supports "decoding". It means that the packed binary can be decoded. As your analysis might be too difficult, you may use the decoding function. (How to decode?

Please figure it out yourself. Allowing the decoding functionality is already a huge compromise.) ***However, you can only use the information from the decoded binary to help build your pintool. Your pintool should be able to handle the unpacked binary and your report should not discuss with the unpacked binary at all.***

2. Do not change the code itself. You are only allowed to change data.

Guide:

1. You can use UPX to pack your own sample to understand what UPX does and how.
2. Logging instructions generated by UPX would be useful in identifying the original binary's instructions.
3. Focus on branches and their conditions.
4. Focus on branches that are not taken.
5. When the condition meets, you will see different outputs. The output (in the console) will have one empty line before the message (i.e., the "This program is legitimate.\n - Source: Dude-Trust-Me.\n").

What to submit?

1. Your Pintool code. (**Submit a single .cpp file please**)
2. A report that includes

(1) high-level descriptions of how your pintool works (around, **0.5 page**) (15%),

(2) what are the triggering conditions that the sample expects (there are 4 conditions) (around **0.5 page**) (25%),

(3) what is the suspicious activity that the sample would do (hint: it will do some operations related to files) (around **1 page** with the details such as instructions for the activities) (40%),

(4) how did you deal with the packer (around **0.2 page**) (5%),

(5) how did you analyze the sample (around **0.5 page**) (15%).

- Please submit only two files: **(1) .cpp file**, and **(2) .pdf file**.

Extra credit:

In practice, you would not be able to use the unpacker. This essentially means that you must deal with a binary with no symbols at all. If you have not used the unpacker, please elaborate what were the challenges you encounter, with concrete examples of challenges you encounter during your analysis. This extra credit should be 0.5~1 page long. The interpretation of the answer and amount of the extra credit is subjective, meaning that you may not receive a credit if your description does not exhibit clear understanding of the task.