Project 2.5 - D&R (Doors and Rooms) Malware

As an optional project, you will have to trigger some malware knowing limited information beforehand. In the real world, you as a malware analyst have to figure out the possible triggers on your own. However, given your brief experience with Project 2, it is difficult for you to determine all triggers a malware might have.

Because of this, we have developed a sort of game for you to play with this final piece of malware. You will be given a single hint that will allow you to solve and find more clues and puzzles in order to successfully trigger the malware. It is up to you to reason about the malware's activity and how to trigger the other behaviors. Don't worry, the triggers you will have to invoke are the same ones you have learned about in Project 2.

Your target malware: http://evan.gtisc.gatech.edu/cs6035/project2.5/infected.zip

Inside the Ubuntu VM from Project 2, unzip the file to find "TriggerMeTimbers.exe" (the password to unzip this file is "infected" without quotations).

First and only hint:

- The malware triggers the first clue on some day in April 2016.

Submission site page: http://malaise.gtisc.gatech.edu:3000/

- Use the same username/password credentials as Project 2

Total points: 100

Part I (50 points: no partial credit)

Using the same virtual machine environment and software and setup from Project 2, find the right combination of conditional triggers to force the malware to show its hidden activities. The solution will be 4 plaintext messages (English words). Don't forget to run "config.sh" if you've shutdown or restarted the Project 2 VM before running Cuckoo.

Part II (50 points: partial credit is possible)

Run TriggerMeTimbers.exe for at least 180 seconds (fully triggered). Note: it may be helpful to re-generate reports from Project 2 using the same amount of execution time for each malware as you use to execute TriggerMeTimbers, but this isn't necessary. Then, group this malware sample (TriggerMeTimbers) along with the other 7 malware samples from Project 2 (malware1, malware2, malware3, malware4, malware8, malware9, malware10) based on their common activities into **3 different groups. A sample can only belong to one group; the same sample cannot belong to more than one group.**

To do this you can do either manual (eyeballing it) or automatic (machine learning) to group these malware samples based on their common behaviors. As you noticed from Project 2, each samples has their own set of unique behaviors in Phase II. Based on these **high-level**

behaviors alone, it would seem as if none of these samples share any significant common features.

Instead of grouping based on these sets of behaviors (which were crafted to be mostly unique to each sample of malware), group based on what processes were created by the malware, what common sequence of API and system calls were made during each malware's execution, what files/registry keys were touched by the malware, strings contained in the executables, libraries imported, etc. Using these **low-level** behaviors, there will be obvious differences in executions in most of the samples (so manual analysis is feasible).

One sample (of the 8) in particular is a bit tricky to group into one of two groups (intentionally). Take what you believe to be the majority of the common behaviors (or features of the malware executable/script itself) and use this to group this last sample.