CS4238 Course Project: Malware Analysis

Due Date: Sunday, 5 May 2024, 23:59 SGT

1 Instructions

This is a **group/team assignment**. You need to work with your teammates, and submit just **one report** for your team. Note that your report may be checked against anti-plagiarism services. Please prepare your report in **PDF format** by using your team number as part of the file name. Upload your report to Canvas's Course Project submission folder. Note that your report should also list the name, student number, and email address of all team members at the beginning. Also, briefly mention **the contributions of each team member** in your group for possible marking adjustments in serious cases of contribution disputes.

2 Assignment Mission

pw: infected

2.1 Learning Objective

The **learning objective** of this project is for your team to analyze a real-world Windows malware sample using techniques that are discussed in lectures and labs. The **following techniques** are thus relevant to be applied on the given malware sample to examine its behavior in Windows OS:

- Basic static analysis
- Basic dynamic analysis
- Advanced static analysis, e.g. using IDA Pro
- Advanced dynamic analysis, e.g using a Windows debugger such as OllyDbg

To help you with the tasks given below, some **sample malware analysis reports** are included with this assignment brief for your reference regarding the report structure.

2.2 Provided Malware Sample

You are given a folder with a sample of `PlugX` malware that was supposedly responsible for the SingHealth cybersecurity incident a few years ago. In this folder, there are two sub-folders that contain the two versions of the PlugX malware, (1) `Inactive-Version` and (2) `Active-Version`.

Inactive Malware The inactive version is a modified version of the malware with the first few instructions changed to execute the `ret` instruction to return/exit immediately after starting execution. This version is intended to be used for Task 1: static analysis (in Section 3.1).

Active Malware On the other hand, the active version is the live malware that is **dangerous** and requires execution in a safe environment as mentioned in the lecture. The active version is intended to be used only for Task 2: dynamic analysis (in Section 3.2). Please **only run and dynamically-analyze** the malware samples for your Task 2, and be careful in doing so in order to prevent them from escaping into the wild!! The samples have an extension `.bin`. To run the sample, first rename them to "`.exe`" extension/suffix, and then run it as an administrator.

3 Assignment Tasks

This course-project is evaluated for **30 marks**, which is worth **30%** of your final marks. Unless there is a significant dispute about your team's member contributions, every team member gets the same marks.

Complete the given tasks and write up your report about the malware behaviour by:

- explaining your analysis steps;
- mentioning the employed tools;
- enclosing necessary **color** screenshots **from your own analysis** (i.e., not from the Internet or from other sources!);
- succinctly highlighting important findings;
- drawing conclusions based on the findings.

3.1 Task 1: Static Analysis of Malware Sample (12 marks)

Perform static analysis of the given malware sample by considering the features mentioned in the lecture. Additionally, do mention the key observations by analyzing the sample. You can include details about the malware such as its payload characteristics and any other interesting or distinctive properties/signatures. Please include details of your analysis in the report, enclosing screenshots when necessary. Follow the guidelines in Section 3 for writing the report.

3.2 Task 2: Dynamic Analysis of Malware Sample (18 marks)

Perform a dynamic analysis of the given malware sample. You may consider performing the following investigation points, but we encourage you to explore additional interesting tasks beyond them:

- Network analysis (using Wireshark): e.g. are there any network-behavior patterns?
- Filesystem behavior analysis: e.g. what files do the malware create/modify/delete? Is there any sequential order/pattern?
- Co-analysis with IDA: e.g. locate the code's functions related to your findings on malware behavior, and make further exploration on these functions.
- Identify the malware's payload and analyze it.
- Any other interesting or distinctive behavior of the malware.

Please write your report based on your analysis, following the guidelines in Section 3.

Good luck, and have fun with your course project!

— End of Brief —