FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Review report of a final thesis

Reviewer:	Ing. Miroslav Prágl, MBA
Student:	Bc. Artem Ustynov
Thesis title:	Light-Weight Sandbox for Installers
Branch / specialization:	Computer Security
Created on:	30 May 2022

Evaluation criteria

1. Fulfillment of the assignment

▸ [1] **assignment fulfilled**
[2] assignment fulfilled with minor objections
[3] assignment fulfilled with major objections
[4] assignment not fulfilled

Student successfully created working (in proof of concept state) software based on assignment and former research and proved it working and usable. I've personally tested it and found it working as expected. It's somehow limited / tailored to certain installer, but there's potential of wider usability - hopefully student will discuss related questions during defense.

2. Main written part

70 / 100 (C)

Main part is sufficient with no noticeable errors / inaccuracies, but there's still a lot to be improved. In some parts the work is way too detailed and "stuffed" with pictures and charts and also somehow hard to read, namely initial research part. Hopefully final discussion and conclusion are clear and make nice resume. Grammar mistakes and typos are present, but these don't have major impact on readability. Student chosen pretty low-level sandboxing level / API hooking and covered / intercepted most common calls, providing another nice space for discussion. External sources are well cited and the software part is clearly made without "borrowing" foreign code.

3. Non-written part, attachments

85 / 100 (B)

As the software is intended for specific / testing purposes and its principles are described in written part, it's quite adequate and does it's job. Code is readable and sufficiently commented. The platform choice was quite predetermined by low OS API level of the project. I was particularly pleased by choosing SQLite as registry redirection target, providing functionality comparable to native Windows registry virtualization. Experiment

is repeatable and software can be possibly extended to cover wider area of sandboxing if necessary,

4. Evaluation of results, publication outputs and awards 80 /100 (B)

Chosen approach and resulting software can definitely be used in practice. If author manages to cover wider area of installers with backend / prepare some intuitive frontend, it could became highly appreciated tool for administrators and skilled users.

The overall evaluation 70 /100 (C)

It's a pity the topic was not worked out to it's full potential, namely the written part. On the other hand I appreciate following:
- The topic of misbehaving or even malicious installers is evergreen in Windows ecosystem, but existing free tools are scarce and/or only partially covering the needs. This project tries to fill the gap.
- Author chosen challenging level of sandboxing where proper documentation / examples are hard to find and lot of individual work is required.
- The project has nice potential to sandbox more than just specific installer, probably up to level of portable container of an application.

Questions for the defense

How can targeted installer detect and possibly jailbreak the sandbox?
Would it be possible (what would it require?) to use your application for sandboxing other installers or more generally speaking other processes?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.