FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Supervisor's statement of a final thesis

Supervisor:	Ing. Josef Kokeš
Student:	Bc. Artem Ustynov
Thesis title:	Light-Weight Sandbox for Installers
Branch / specialization:	Computer Security
Created on:	16 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

The assignment was completed. The student analyzed the subject field and proposed a technical solution to sandboxing an installer, then went on to implement a proof of code of such a sandbox.

2. Main written part 70/100 (C)

The written part of the thesis is somewhat unbalanced. While many parts are quite detailed and deal with the topic at a level beyond expectations, others can be more problematic. In particular, I am not too happy about the initial analysis (chapters 3 and 4) which seems correct but rather disorganized (e.g. section 3.1 really should be section 3.4 or later) and I feel a reader needs to be fairly familiar with some of the concepts involved to fully understand the content of the work.

I am especially worried about the fact that the core of the analysis and the foundation of all of the implementation is based on the observed behavior of a particular installer rather than on an analysis of what could possibly be done by a rogue application. As a result, I can propose several attack vectors that would escape the application's notice, such as simply using a GetModuleHandle and GetProcAddress to find the actual address of an otherwise sandboxed function. I feel this is caused by the student's narrow focus on a particular use case, i.e. a specific installer based on the InnoSetup engine.

On the other hand, I really like the work's focus on intercepting the functions in NTDLL rather than on the user level - despite the limitations and issues inherent in this approach (which are described in the thesis), this is an area that's not well studied and documented and any insight into is is useful.

The language level of the work is acceptable. I did notice a number of usual errors (e.g. the use of articles) and some errors that shouldn't be there (incorrect words such as "crush" or "instinctive" instead of "crash" or "incentive"), as well as some typographical issues (e.g. the English abstract split over two pages, some incorrect paragraph boundaries etc.), but they don't prevent the reader from understanding the work.

3. Non-written part, attachments 70/100 (C)

The provided code is a proof of concept of the student's proposed sandbox solution.

As far as the code quality is concerned, it is clean and reasonably easy to understand, although it could be updated for a better readability by providing more comments (in the code, comments in the headers are fine) or using a more consistent formatting (e.g. use curly braces around all blocks, not just some, use blank lines to separate blocks of code). I am not convinced that using exit() when an unexpected condition was encountered is the right thing to do, either. The directory for the captured data needs to be much more carefully managed, currently some accesses depend on the real-time value of the working directory while others don't, which can easily cause files to appear all over the filesystem if the sandboxed application changes CWD frequently.

Regarding the functionality, the code works fine if the assumptions from the analysis hold. That, however, is a significant IF - I did encounter serious problems when trying to use the application both outside the expected scope and when trying to keep within it. Even some provided test cases, particularly those dealing with files, failed, and I wasn't able to capture anything from Far Manager (the injector failed to inject the sandbox DLL). While I am ready to admit that both my system and the tested applications are far from what the student expected to encounter, I feel that the application should be able to gracefully handle these cases, or at least report what the problems are. Still, it's OK for a proof of concept.

4. Evaluation of results, publication outputs and awards 60/100 (D)

In its current form, the thesis is a study of a concept rather than an immediately applicable work. Some aspects are easy to fix in the future, e.g. achieving compatibility against more and more varied systems, others would be more difficult to overcome. In particular, I am convinced that a more systematic analysis is needed, built around the functions available to the target applications rather than around the specific behavior of one of them. I also feel that the student would do well to widen his outlook - while his focus was on installers, this tool, when finished, could be very useful for non-installers as well, as a means of running applications in a "portable mode" even though they were not designed that way.

5. Activity of the student

▸ **[1] excellent activity**

[2] very good activity

[3] average activity

[4] weaker, but still sufficient activity

[5] insufficient activity

6. Self-reliance of the student

[1] excellent self-reliance
▸ **[2] very good self-reliance**
[3] average self-reliance
[4] weaker, but still sufficient self-reliance
[5] insufficient self-reliance

The student is self-reliant, but does need some guidance in order to work on the right things.

The overall evaluation 70 /100 (C)

While the work lacks in certain areas, most pressingly in the analysis of the subject area, the organization of the text and the compatibility of the created code, it is by no means bad. The student selected a topic which can seem simple on the surface but is quite challenging in reality, especially when he decided to perform the sandboxing on the level of NTDLL rather than the well-documented user-level libraries, and handled it quite well. The complaints I have go mostly towards the reliability of the product rather than the concept itself, and that is a matter of evolution of the code rather than getting it perfect on the first try. While the created tool doesn't perform as well as I had hoped for, I still think the work is Good - and that's the grade I propose.

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.

Activity of the student

From your experience with the course of the work on the thesis and its outcome, review the student's activity while working on the thesis, his/her punctuality when meeting the deadlines and whether he/she consulted you as he/she went along and also, whether he/she was well prepared for these consultations.

Self-reliance of the student

From your experience with the course of the work on the thesis and its outcome, assess the student's ability to develop independent creative work.

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.