

# Web Security and Malware Analysis

## Assignment 9 - 18-20/05/2020

**Goal:** Solve each task by providing a brief explanation of the solution that you adopted. You must use IDA PRO (for disassembling/debugging) and GHIDRA (for decompiling), or one of the indicated tools to solve the tasks (when required). Explanations should mainly include screenshots and some brief comments.

Note: Although you may find the solutions to many of the proposed tasks on the web, *try to solve them on your own and do not immediately give up*. This training will be very useful for the practical question that will be asked during the exam. Also, remember to give **brief written** answers (avoid copy-and-paste from other tutorials and try to write your own answers) and to use screenshots to describe what you do. You are also recommended not to use copy-and-paste texts from tutorials, but to use your own words.

**Deadline:** All assignments must be sent before taking the final exam.

**What to do:** Send the report in the PDF format to [davide.maiorca@unica.it](mailto:davide.maiorca@unica.it) with the subject “[WEBSEC] - Report For Assignment 9 - NAME SURNAME” and name the file “websec\_report\_9\_name\_surname.pdf”. Remember to include your name, surname, and matriculation number in your report!

**Starting Notes:** Please remember that the files you are going to deal with are real malicious samples, which could potentially destroy your machine. DO NOT EXECUTE OR OPEN THEM in your own system. ALWAYS USE A VIRTUAL MACHINE (VMware/VirtualBox) FOR EVERY OPERATION. Save a snapshot. Additionally, you could execute the tools in a virtualized environment with the

Sandboxie tool (<https://www.sandboxie.com/>). You find the executables for this assignment on <https://gofile.io/d/EyHLSW> (password: malwanalysis)

**Additional notes:** if you have problems with Windows Defender, please follow these instructions to disable it:

<https://www.windowscentral.com/how-permanently-disable-windows-defender-antivirus-windows-10>

If you have problems at running IDA PRO, please install Microsoft Visual C++ Redistributable Package:

<https://www.microsoft.com/en-us/download/details.aspx?id=5555>

To use GHIDRA, you have to install the Java JDK at the following link:<https://www.oracle.com/java/technologies/javase-jdk14-downloads.html>

## Task 1 - Full Malware Analysis

You are required to perform a full malware analysis of the sample Lab09-02.exe.

In your answer, **you must employ all the techniques learned so far, in particular:**

- Analyze the PE structure;
- Take a look at the employed strings;
- Use general static analysis with IDA/Ghidra;
- Perform dynamic analysis with off-the-shelf tools (procmon, regshot, process explorer...);
- Perform dynamic analysis with IDA (or other free debuggers as OllyDbg).

You are free to go in depth as much as you can, **but be sure to motivate at least some malware behaviour by clearly showing how you obtained it.**

**Remember to always take snapshots of your VM before the execution. You will most likely need to restore your VM.**

As a general hint, this sample is easier to analyze than the one of Assignment 8 - Task 1.

Note: if you have problems using IDA Pro 7.0 for debugging 32 bit executables, here is what you can do:

1. Switch to IDA PRO 6.8
2. Use IDA Freeware ([https://www.hex-rays.com/products/ida/support/download\\_freeware/](https://www.hex-rays.com/products/ida/support/download_freeware/))
3. Used OllyDBG (<http://www.ollydbg.de/> - You can find a guide for OllyDBG in Chapter 9 of the “Practical Malware Analysis” book.

## **Task 2 - Windows API Functionality**

You are required to perform a full malware analysis of the sample Lab07-01.exe. You can either use static or dynamic analysis.

Your goal is to describe the malware behaviour by pointing out the following Windows API-related structures (if present):

- Mutex
- Threads
- Services
- Communication API

## **Task 3 - More Static Analysis**

Given the assembly X86 code contained in the file assignment\_9.txt, answer the following questions:

1. Describe what each part of the code (starting with LOC...) does. As usual, you are not required to analyze every single assembly instruction.
2. Describe the functionality of the code and write the equivalent program in C or in a pseudo-language. What does the program print?

