LINGI2144 – Assignment 2

April 2020

This assignment is to be completed in **groups of 2**. Note that for organisational reasons you must be in **the same groups** as you were for first assignment. Do no hesitate to use the Virtual Machine used for the labs (the binaries have been compiled on this machine).

Due date: **2021-05-13 18h**

This assignment has 2 parts. Each part count for 50% of the assignment's mark.

1 Static Malware Detection with YARA

Your goal for this part is to build YARA rules in order to detect malware. We consider here that malware is any program that does or at least tries to do one of the following:

- print `"Give me bitcoins!"`
- write to `"/etc/sudoers"`
- delete `"/root/secret"`

Your YARA rules will be tested against samples of malware and cleanware. We do not provide any sample program (malware or cleanware) for this part, but you are encouraged to write your own.

Marking will be based on true and false results. Like in the tutorials, your rules are not expected to be 100% accurate, so do not worry if you cannot detect every malware (or you misclassify some cleanware). The goal is to have a good rule, not a perfect rule.

2 Reversing malwares

For this part of the assignment, your goal is to revert the actions of a malware that encryted all your files in a certain directory.

Warning: these malware programs will encrypt all files in the directory they are run in. They will not delete any files, but since they make copies you should be careful about running them in directories with very large files. Do no hesitate to use the Virtual Machine used for the labs (the binaries have been compiled on this machine).

You have two options here. You only have to do one of both options.

Option A - Dynamic Analysis of malwares

For this part, we provide you with two malware programs (`malware_1`, `malware_2`) that encrypt files in the directory they are run in. Both malwares are using a simple and reversible encryption that only take one instruction (input XOR key and input - key). Your goal is to:

- determine the key used for encryption;
- determine the size of the key used for encryption;
- determine the method used for encryption (which malware is using the XOR and which malware is using the substraction);
- explain how you found this key and how your could decrypt an encrypted file.

You can run the malware programs with the command `malware_X <name>`, where `name` is your name and is a single word, without spaces, using only the characters a to Z. This argument will relate to the key generated. When submitting your assignment, **state which name was used with the key you found**.

Warning: before starting, **on the virtual machine used for the labs**, in order to install the additional library needed for this exercice, run the following commands on the virtual machine:

```
wget http://old.kali.org/kali/pool/main/o/openssl/libssl-dev_1.1.1d-2_i386.deb
sudo dpkg -i libssl-dev_1.1.1d-2_i386.deb
```

2.1 Hints

- These operations can be reversed by applying the inverse operation.
- There are several methods to solve this problem: either look at the instructions and find where the encryption is done or when you have found the key, compare the result before and after the encryption of your files.

Option B - Binary Analysis and exploitation

For this part you are provided with a malware (`malware_3`) that, similarly to the ones in Option A, encrypts the files of the current directory. By looking at the code you realize that the author of the malware left the code of the decryption function. You also find that the code is vulnerable to a format string attack that could overwite a variable and trigger the decryption. Your task is to trigger that decryption function contained in the binary to decrypt the encrypted files.

You can run the malware program with the command `malware_3 <name>`, where `name` is your name and is a single word, without spaces.

Your answer must contain:

- your understanding of the execution flow of the program and the commands you used to understand it;
- the input you are using to trigger the decryption of the files;
- in general terms: how the format string attack could have been averted.

Warning: Your solution to trigger the decryption function must work outside of external tools like GDB. You are free to use GDB and other tools to analyse the binary, but your solution must be an input value given to the program.

Deliverable

This project must be done in groups of 2 students. Note that for organisational reasons you must be in **the same groups** as you were for first assignment.

The group is tasked to provide a zip file containing:

- A PDF containing the answers and detailed explanations for the problems;
- If applicable: any source code needed to reproduce your findings;
- Any other material that you deem relevant to understand your solution.

The deadline for this project is on the 13th of May 18h. You will be able to submit your solutions for the project on the moodle page of the course.