BLG460E - Secure Programming

Take-Home Exam 3 : Reverse Engineering

Due Date: 19 April, 2019, 23:59 PM

This assignment aims to provide hands-on experience on reverse engineering ELF files. In the assignment there are 6 executable files you should analyze. You are required to obtain as much information as possible and prepare a report that contains your interpretations on the information you gathered from the files. The files are zipped in a file. The password is **secprog3** for the zip file.

Part 1: ELF file structure

Study on Executable and Linkable Format (ELF) file structures. Provide the parts of an ELF file in the report (headers, sections, etc.) in general and make comment on each part. Note that you are not expected to mention all fields in the parts, only big parts will be enough. Do not copy-paste from the internet. Try to use your own words.

Part 2: Analyze the given files

When you unzip the given file you will see 6 files. The first four files will be used in this part of the assignment: **file1, file2, file3, file4**. Analyze the files and answer the following questions for each file in the report.

- Is the file malicious or benign? Explain how you come to that conclusion.
- What is the entropy of the file? What can you say about the files when you compare their entropies?
- What are the basic information about the file (filetype, machine architecture, etc.)? How did you obtain these information?
- Does the file contain any debugging information or not? How did you find out this information?
- Can you disassemble the file? Explain why you cannot or how you can.
- When you inspect the strings in the binary file what information can you say about the file? What are the capabilities of the file? Here are some examples:
  - The file may be gathering information from /proc/ folder.
  - The file may be making HTTP requests.
  - The file may be running a shell command.
  - The file may contain an IP address.
  - The file may be manipulating the processes.
  - The file may be making some file operations.

  Explain how did you conclude your answers.

- Try to run the file on a virtual machine. Were you be able to run the file? If it worked, what happened? If it did not work, why?

  You can use the Ubuntu virtual machine from the first assignment. You can download it from `https://drive.google.com/file/d/16Am5PDT4fcEkhMBjIqKVToC5gbGT9wJx/view?usp=sharing`. Login password: **12345**.

- Is the file packed or not? Explain how did you find out whether it is packed or not. If it is packed try to unpack the file and answer all the questions above again for that file. Compare the results. Make comment on the differences.

- Considering the answers for the questions above, what is the purpose of the file? What does it do?

Part 3: NO DEBUGGING!

Analyze the file named **file5**. Try to run the file directly on Linux. Then, run it using `gdb`. What are the results?

Try to find a way to inspect the content of the file. When you understand the problem, run the file properly. Explain what the program does and how did you solve the problem.

Hints: Which functions are used in the file? What are the strings in the binary file? Can you disassemble the file?

Part 4: Broken file

Analyze the file named **file6**. The file seems broken. Try to find the problem and fix it. Then, run the file again. Explain the reason of the problem and what the program does.

Hints

You can use any program to examine the files. Here are some programs that can help you with that: **ent, nm, radare2, ltrace, strace, file, strings, gdb, objdump, hexdump, readelf.**