ESPLab B 2022

Lab 9

Lab 9: ELF - Virus (Architecture and SPlab)

Lab Goal:

In the previews lab, You have learned how to manipulate ELF files by writing a simplified version of the readelf program. In this lab, You are requested to write and run Proto program which will attach itself to executable files and infect them.

Note that all the functions in this lab will be implemented in assembly language

Task 0:

ELFexec1 and ELFexec2 files are almost identical except that one of them is **infected**,

Task 0A:

Download both files and Answer the following questions(be prepared to explain your answers to the lab instructor):

- Where is the entry point specified, and what is its value?
- What fields inside the file header differ? what does this mean?
- Where in the file does the code of function "main" start?
- Which file is infected?

Task 0B:

In this task, you are required to fix the infected file manually using Hexedit and readelf

Task 1: First Step towards proto-virus

Many computer viruses attach themselves to executable files that may be part of legitimate programs. If a user attempts to launch an infected program, the virus code is executed before the infected program code. The goal is to write a program that attaches its code to files in the current directory.

In the following task, You will implement base functions for the proto program.

Use skeleton.s as a skeleton file.

Task 1A:

In this task, you are required to edit the skeleton.s file to check if the file associated with the FileName label is an ELF file, if so print the content of the OutStr label otherwise, print the Failstr label.

Task 1B:

In this task, We will start the infection process by adding the code that prints OutStr to the end of the Target file(FileName) if the target file is an ELF file.

Task 2: Proto-virus

In the following tasks, You are required to Implement the ***Proto-virus*** program.
Note that you will need to reuse the code from Task 1 in order to complete this task.

Before you start your implementation!

- How could We run our virus after attaching it to the original program?
- How could We run our virus without affecting the original program run?
- **Hint:** How changing the entry point could affect our run?

Task 2A:

In this task, you'll need to extend your code from Task 1B, which enables the virus to infect other executable files.
Similar to what we did in Lab 4, running the proto-virus program will copy its own code (the virus) to the end of the ELF file (FileName). Unlike Lab 4, we'll make the copied code run whenever the infected file is executed. To do so, after attaching your code to the ELF file, You are required to modify the ELF file to enable it to print the associated string(Virus).

Task 2B:

As mentioned above, computer viruses attach themselves to executable files. If a user attempts to launch an infected program, the virus code is executed before the infected program code. But it should not affect the original program run and the program should run smoothly.

In this task, you are required to extend your implementation of the *Proto-virus* program to resume the execution of the original program after running the code of the virus. To do so, you'll need to do the following:

1. Before infecting the file, save the previous entry point.
2. After infecting the file, copy the previous entry point to the code of the virus (where? You'll need to figure this out)
3. Add additional code to the virus that will enable it to resume the execution of the original program by jumping to the previous entry point after executing its own code.

Deliverables:

Tasks until 2a must be completed during the regular lab. Task 2b may be done in a completion lab, but only if you run out of time during the regular lab. The deliverables must be submitted by the end of the lab session.
You must submit source files for task 1b task 2a and task 2b and also a makefile that compiles them. The source files must be named **task1b.s task2a.s**, **taskbc.s,** and the makefiles named **makefile1b**, **makefile2b,** and **makefile2c**.

Submission instructions

- Create a zip file (the name of the file should be your id) with the relevant files (only).
- Upload the zip file to the 'submission of Lab 9' bullet in the moodle.
- Download the zip file from the moodle and extract its content to an empty folder.
- Compile and test the code to make sure that it still works.

Last modified: Monday, 13 June 2022, 7:57 PM

Administration

› Course administration

◄ Submission of Lab 8

Jump to...