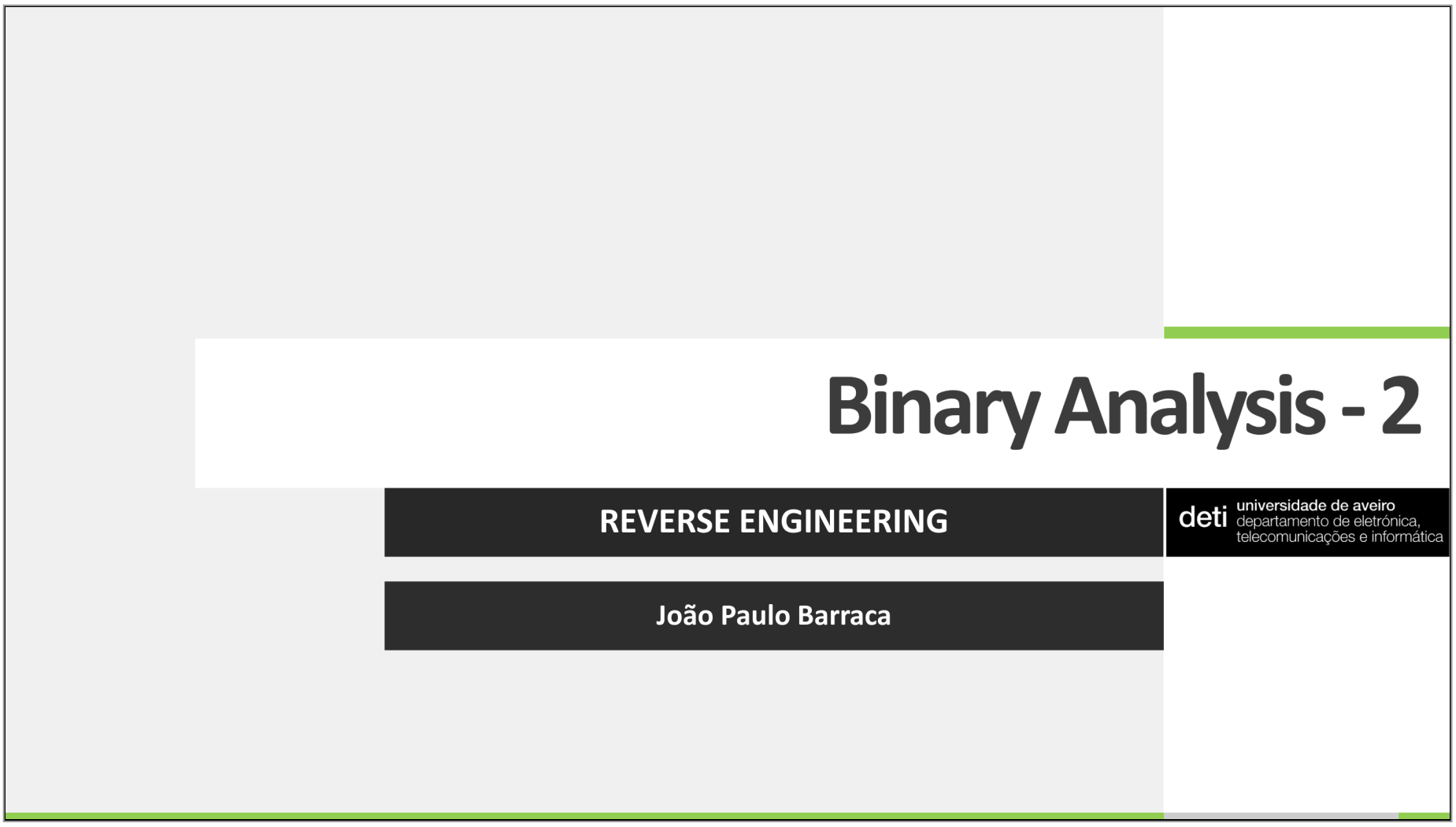
Static Analysis of Applications

Lecture Notes

Analyzing Binary Executable files such as ELF. Focus on static analysis, calling conventions and decompilation.

Previous Next



Download here

Practical Tasks

Exercise 1

Consider this binary file (from the HTB Cyber Apocalypse 2021 CTF), and use it to explore ghidra. Rename functions and variables, and set proper types. It helps if you check the manual of the C functions presented there, as you can see what is the actual type (although ghidra should do it).

This is also a good example to demonstrate the use of plugins. One of the plugins in ghidra, which can be used to `XOR` memory. Use it to decrypt the password and obtain the flag. You just need to select the memory to `XOR` and run the plugin.

Exercise 2

Use the provided samples and examine the calling conventions and their impact to the code produced. For each sample, compile it and analyze the assembly produced. You can compile the samples with the included `Makefile`, which will automatically produce both `ELF` and `Assembly` files for each convention.

Exercise 3

There are several crackmes available for you to improve your reverse engineering skills. For each one, write a small text with the conclusions obtained regarding its behavior. If you can, create an equivalent program in another programming language such as `C` or `python`.

The objective for all these crackmes is to find a code or text, and get the `Correct` string printed to the standard output..

Follow a structured approach to these programs:

- Identify the main structure of the program (functions)
- Go through the code and rename variables and functions to match their purpose
- Identify building blocks inside functions.
- Create representations of part of the program in another programming language (e.g, python).
- Document everything for future discussion.

Along your analysis, take nodes of your assumptions. Take in consideration that a Reverse Engineering tasks don't produce the original design, but a limited and potentially flawed interpretation of it.

Exercise 4

This unknown file has no structure and we have no information about its architecture or purpose. What information can you extract from it?

Analyze it and discuss your ideas with the class.

With the contents already studied during these classes (static analysis), it is not possible to fully access the file and obtain its content. Still, it is a valid file for some format (`file`?? `binwalk`?? `qemu` ?? `PDF reader`), and shows how important it is to have some basic information about the architecture and structure. We will get back to it in the future.

Tools and links

- ghidra: https://ghidra-sre.org/
- objdump: https://man7.org/linux/man-pages/man1/objdump.1.html
- readelf: https://man7.org/linux/man-pages/man1/readelf.1.html
- LIEF: https://lief-project.github.io/
- gnutls:https://gnutls.org/documentation.html
- HxD: https://mh-nexus.de/en/hxd/
- bvi: http://bvi.sourceforge.net/
- ImHex: https://github.com/WerWolv/ImHex
- HexWorkshop: http://www.hexworkshop.com/
- ghex: https://wiki.gnome.org/Apps/Ghex
- HexEdit: https://hexed.it/
- FileInsight: https://github.com/nmantani/FileInsight-plugins

PREVIOUS
Binary Executable Files

NEXT
Project 1 - Android Reversing

Last updated on 21 Mar 2024