# CCF Lab 3 - Sandboxing & Malware Analysis

In this lab, you will get the skills to work with sandboxes. A sandbox is a limited environment on your system for executing guest programs without access to the main operating system. It is a mechanism for the safe execution of programs. Sandboxes are often used to run untested code, unverified code from unknown sources, and to run and detect malware. After that, you will also learn how to detect malicious code artifacts using static analysis.

## Task 1 - Preparation

1. Prepare and install a sandboxing solution that allows you to configure the sandbox environment LOCALLY. For example, **Cuckoo**.
2. Use any virtualization environment, better to use the latest version (check repo and official website).
3. Create a Virtual Machine and set up it with your sandboxing solution. Make sure that VM uses a **HOST ONLY** network adapter.

## Task 2 - Let's get some malware

1. Download some malware/ransomware from the Internet (for example, TheZoo repo). Please be careful when you run them, **THESE ARE REAL MALWARE**.
2. Select at least two malware that you want to analyse in the sandbox VM that you prepared in Task 1.

## Task 3 - Sandbox Analysis

1. See what kind of traces, artifacts, and connections your sandbox VM detects.
2. Analyze the behavior of the malware, and then write about what the malware does and what its goal is.
3. Does the malware have some sandbox detection? If yes, try to defeat/detect the techniques that are used for that.
4. Extract a memory dump of the sandbox analysis and analyze it using **Volatility** or any other supported tool. Can you trace some of Cuckoo's findings in the dump? (*write any other potential IoC you find*)

5. Try to use other online tools (*for example, any.run , hybrid analysis*, ...), figure out if these online platforms will manage to detect more artifacts than what you have found.

## Task 4 - Static Analysis

1. Use any tool for static analysis of your selected malware (*for example, Ghidra, IDA, Binary Ninja, Hopper, Radare2*, ...).
2. Try to see if there are some artifacts that dynamic analysis did not manage to find, for example a piece of code that did not run inside the sandbox VM.
3. Try to describe which method is better (Sandboxing V.S. Static analysis) is better, and which one is more useful in which case.

## Bonus - Dynamic Analysis

1. Try to use some debugger to analyse the malware **WHILE IT IS RUNNING**, be careful where you will run this malware, the debugger that you select must have a remote debugging feature.
2. Try some debugger that will allow you to debug the whole operation system (for example, PyReBox (https://github.com/Cisco-Talos/pyrebox)).
3. What kind of benefit does this method have?