# Lab 1. Program analysis to determine its malicious nature

The goal is gathering evidence to confirm or deny the malicious nature of the analyzed program.

### Sources for identifying programs with questionable reputations

For an experience as close as the real life, it's recommended to look for malicious programs from the following sources:

1. Cracked apps, like: windows/office activator, cracked adobe products, free antivirus, etc;
2. Intrusively promoted apps on websites for: free movies, crypto, bets, dating;
3. Untrusted groups/forums;
4. Own creativity, past experiences, currently used cracked apps.

## Analysis tools

- pe-bear;
- Ghidra;
- ILSpy;
- ProcessMonitor;
- System Informer (former Process Hacker);
- Wireshark;
- Autoruns;
- TCPView;
- Regshot.

## Steps

### 1. Malware analysis environment

1. Download and install virtualbox;
2. Set up the virtual machine:
    1. Download and import the virtual machine (You can follow these steps for vm importing);
    2. Learn about snapshots and why do we need them. Here you have a small and clear article on this subject.
       You should have a snapshot:
    - after import;
    - after installing the required tools you need for analysis;
    - before any run of the malicious program. This allows you to run the program as many times as you need, and every time "on clean", like it never run before on that machine.
3. Install the tools (read until the end to know exactly what you'll need).
4. It's recommended to remove any shared folders and set the network type as NAT.

### 2. Static analysis

At this step the following tools will be used:

- pe-bear
    - Obtain human readable strings. Find imported functions from libraries that are not present into include table of the executable file. Find domain names. Look for suspicious text, like: hacked, ransomware, tor browser, bitcoin;
    - WinAPI functions to predict the behavior;
    - The structure of the binary file. If it has empty holes (parts filled with 0 and not real instructions), it is most likely the program wants to avoid detection by antivirus software.
- ghidra - software reverse engineer tool which can generate a pseudo cod based on the executable file.
- ILSpy - .NET assembly browser and decompiler.

### 3. Dynamic analysis

Don't think you can do this step from the first try. The learning path is made of multiple retries.

Start wireshark, procmon, analyze the data they catch, try to master the filters for your needs. Try saving the data to an offline file (offline - nothing is being analyzed).

Run System Informer, autoruns, TCPView, regshot, play with them, learn what they do.

> Before running any malicious program, don't forget to create a snapshot

> A good idea would be to record the screen during the analysis. VirtualBox has this feature build in.

There are many malware analysis articles online, it's fine to learn from them, but:

> If the final report has results from tools any other than the ones recommended here, it may be considered as cheating!