Model Checking for Malware (Virus) Detection

Directrice de thèse: Tayssir Touili (touili@lipn.fr)
Laboratoire: LIPN

The number of malwares that produced incidents in 2010 is more than 1.5 billion. A malware may bring serious damage, e.g., the worm MyDoom slowed down global internet access by ten percent in 2004. Authorities investigating the 2008 crash of Spanair flight 5022 have discovered a central computer system used to monitor technical problems in the aircraft was infected with malware. Thus, it is crucial to have efficient up-to-date virus detectors. Existing antivirus systems use various detection techniques to identify viruses such as (1) code emulation where the virus is executed in a virtual environnement to get detected; or (2) signature detection, where a signature is a pattern of program code that charaterizes the virus. A file is declared as a virus if it contains a sequence of binary code instructions that matches one of the known signatures. Each virus variant has its corresponding signature. These techniques have some limitations. Indeed, emulation based techniques can only check the program's behavior in a limited time interval. They cannot check what happens after the timeout. Thus, they might miss the viral behavior if it occurs after this time interval. As for signature based systems, it is very easy to virus developers to get around them. It suffices to apply obfuscation techniques to change the structure of the code while keeping the same functionality, so that the new version does not match the known signatures. Obfuscation techniques can consist in inserting dead code, substituting instructions by equivalent ones, etc. Virus writers update their viruses frequently to make them undetectable by these antivirus systems.

To sidestep these limitations, instead of executing the program or making a syntactic check over it, virus detectors need to use analysis techniques that check the *behavior* (not the syntax) of the program in a *static* way, i.e. without executing it. Towards this aim, we propose to use *model-checking* for virus detection. Model-checking is a mathematical formalism that can check whether a system satisfies a given property. It consists in representing the system using a mathematical model M, the property using a formula φ in a given logic, and then checking wether the model M satisfies the formula φ. Model-checking has already been applied for malware detection. However, the existing works have some limitations. Indeed, the specification languages they use have only been applied to specify and detect a particular set of malicious behaviors. They cannot be used to detect *all* viruses behaviors. Thus, one of the main challenges in malware detection is to come up with specification formalisms and detection techniques that are able to specify and detect a larger set of viruses.

The purpose of this thesis is thus to:

1. Define expressive logics that can be used to compactly express malicious behaviors. Our goal is to be able to express a large set of malicious behaviors that were not considered previously.
2. Define efficient model-checking algorithms for these logics.
3. Reduce the malware detection problem to the model-checking problem of these logics.
4. Implement these techniques in a tool for malware detection and apply this tool to detect several malwares.