



UML for Embedded Systems

# Exam FALL 2018
# FortiSandbox Software

Ludovic Apvrille

ludovic.apvrille@telecom-paristech.fr

http://soc.eurecom.fr/UMLEmb/

During an exam, you are not supposed to talk with someone else, by any means (including mobile phones, chat, etc.). Access to Internet is restricted to the website of the UMLEmb course only. You may consult your own UML/SysML models made in the scope of the labs, but not other models. Electronic devices are not allowed at all, apart from the desktop computers of the laboratory room ;-).

A grade is provided for each question. 1 bonus point is awarded for writing quality (report and models).

# 1 Objective

Your objective is to model the **software of a FortiSandbox**.

You have exactly 3 hours to model this system, and answer various questions: the time is very short. This means that **you have to take modeling assumptions**. **Keep your diagrams simple and readable**, in particular the analysis diagrams.

Your grade takes into account your report and your models. At the end of the exam, **reports** (in pdf format) and **models** (in TTool format) **must be sent to me by email**. Also, **the report must be printed** and **given to Alexia Cepero right after the end of the exam session**. The report should contain explanations concerning your models, as well as relevant screen captures of models (e.g., interesting simulation traces, formal verification results).

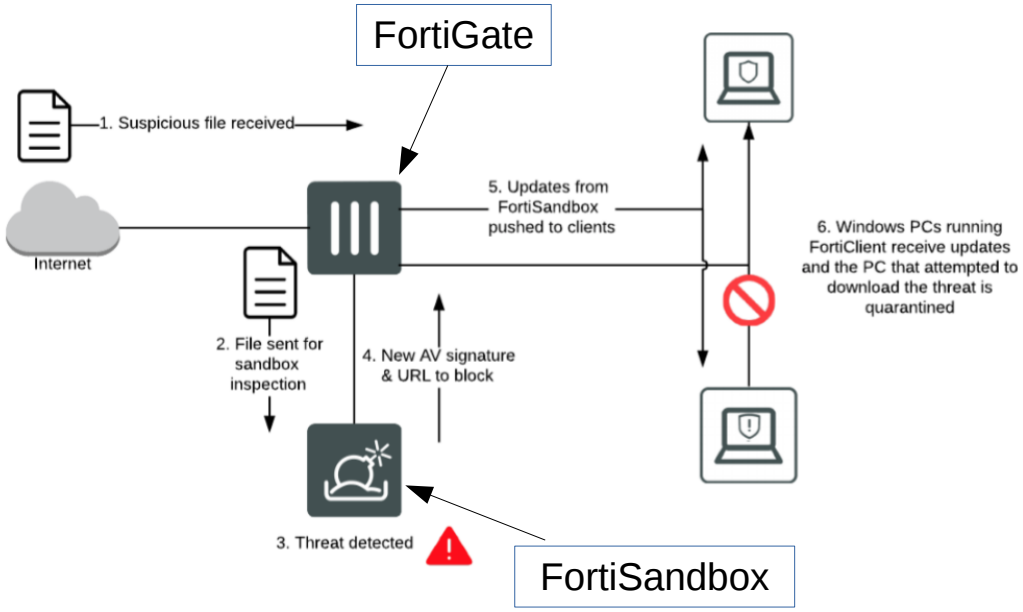
# 2 System specification

Again, the system to model is the software of a FortiSandbox.

## 2.1 Description

### 2.1.1 Overall description

The inspection of files transiting on a network can be done using FortiSandbox. The files themselves are intercepted by a FortiGate and then forwarded to a FortiSandbox that analyses the files in order to detect threats capable of bypassing other security measures, including zero-day threats. The main procedure is given by the Figure just below.




The FortiSandbox uses virtual machines (VMs) running different operating systems to test a file and to determine if it is malicious. If the file exhibits risky behavior, or is found to contain a virus, a new signature can be added to the FortiClient virus signature database. A FortiClient is an Antivirus software running on PCs connected to the same network.
When a FortiGate learns from FortiSandbox that an endpoint is infected, the administrator can quarantine the host, if it is registered to a FortiClient.
FortiSandbox has a VM pool and processes multiple files simultaneously. The amount of time to process a file depends on hardware and the number of sandbox VMs used to scan the file. For example, it can take 60 seconds to five minutes to process a file. FortiSandbox has a robust prefiltering process that, if enabled, reduces the need to inspect every file and reduces processing time.

# 3 Assignments

## I. Assumptions

1. Your assumptions should be clear. Do list them in the report: that list might evolve according to the models you will make afterwards. Make a clear separation between environment and modeling assumptions. [2 points]

## II. Requirements

1. Create a requirement diagram. [3 points]

## III. Analysis

1. Make a use case diagram. [3 points]
2. Continue the analysis in the form you want: activity diagrams, nominal scenario, error scenarios, . . . : you are free to use the diagrams you want. Of course, the idea here is to show important points of the specification. [3 points]

## IV. Design and validation

1. Make a block diagram. Put the emphasis on which blocks are used to model the system being designed, and which ones are used either to model the environment, or to prove properties (observers). [2 points]
2. Draw state machines, and provide a nominal simulation trace, as well as an error trace. [3 points]
3. Prove that whenever an analysis is performed, there is a scan result returned to a FortiGate. Also, from requirements, define a property of your choice, and prove whether it is satisfied (or not!). And obviously, explain how you have modeled those properties [3 points]

**Good luck, have fun!**