Summary

- **Candidate**: Fabio VALLONE (s224870)

- **Title**: Remote Attestation on light VM

- **Supervisor**: prof. Antonio Lioy

- **Accademic Year**: 2016-2017

In the last decade Cloud Computing has massively entered the IT world, changing the way services are offered to the final users. One important aspect of Cloud Computing is the needs to provide greater scalability and flexibility and this can be done by using a technique called Virtualization. This enables us to execute different virtualized ambient on a single piece of hardware. Each virtualized ambient can be seen as a node that offers some services to the final users by exchanging information with other nodes. For this reason the ability to trust each other is growing exponentially. This process is called Remote Attestation and it is from this consideration that the thesis work start.

When we say that a node is trustworthy, we are saying that we are sure that all the file that are loaded into memory are the ones that are supposed to be loaded. In other word we need to be sure that the code, and in general the files loaded on that node, have not been tampered by anyone in order to produce a different behaviour of the node. It is important to note that we are not checking in anyway that the behaviour of the machine is logically correct, but we are simply checking that no one has modified it, so bug and logical error can still occurs in the code. In order to prove its trustworthiness, a single node must be able to report his integrity status to a third one. This can be done by using the techniques and technologies developed by the *Trusted Computing Group* (hereafter TCG). The set of the solutions proposed by the TCG constitute the *Trusted Computing technology*. They are all based on a particular chip called *Trusted Platform Module* (TPM), which is now broadly available in most modern motherboard. Basically the TPM offers two type of operation: Secure data storage and cryptography. In particular the data can be stored in a very reduced set of registers, called PCRs. For the purpose of the Remote Attestation

process, it is possible to store every measure done by the TPM inside the PCRs by simply extending the old value of the register with the new one by using an hash function.

All the techniques developed by the TCG were initially developed in order to work with physical machine. Since nowadays most of the node in the cloud are no longer physical machine, all the concept introduced with the Trusted Computing methodologies have been recently extended to cover also virtual nodes. However in the last years a new virtualization technique, called Light Virtualization, is raising fast. The main advantage of this new technique is that is generally faster and less resource consuming on the host machine. This is done by avoiding to load a full copy of the OS for each virtualized ambient, but instead, using layers and granting direct access to the resource of the machine to the various virtual nodes. In particular, in a standard virtualization mechanism, all the nodes are managed by a broker, called *hypervisor*, while with the light Virtualization, they are managed directly by the host operating system. All the thesis work is focused on a particular light virtualization engine called *Docker*.

Less isolation means also less security. For that reasons we need to extended the work done by the TCG also to the light virtualization world. Some work in that direction has already been done by the TORSEC research group inside the Politecnico di Torino. They proposed a solution that makes use of a modified version of the *Open Attestation Framework* (OAT) and the IMA modules, in order to create an architecture that is able to attestate an host running on docker. This solution exploit the IMA module of the kernel in order to create a list of measurements associated with each container running inside an host. By exploiting the OAT framework it is possible for an Appraiser, to send an *Attestation Request* to a node, that respond with an *Integrity Report* (compiled using the list of measurement done by IMA). The IR is than checked by controlling that all the hashes are available also in a whitelist db. If some of the hashes into the IR are not contained into the whitelist db, then the container is probably under attack.

However the solution is based on a functionality available inside the docker storage driver *DeviceMapper* that has been extensively changed starting from v1.10. The thesis work starts by adapting the existing architecture to the latest version of docker. This is done by studying how DeviceMapper is changed after v1.10 and how it maps each container to the underlying devicemapper folder. After a brief analysis, a new logical mapping procedure has been developed. Though the solution is working, it also introduces a big overhead in terms of time when creating an Integrity Report. This big overhead comes from the extensive use of I/O operations in order to retrieve all the information necessary for the mapping. Since a big part of them are already available in the docker CLI, and the GO programming language is highly optimized for managing the Unix sys call, all the mapping logic is moved

to a new docker CLI command, called *raInfo*.

After resolving thie initial issue, the work moves on by identifying some flaws of the architecture proposed and how it is possible to resolve them. In particular it was necessary to study how IMA works in detail, since, by the default, a particular file is measured only the first time it is actually loaded into memory. Since with docker is possible to share a portion of the host filesystem between multiple container, it is also possible that a particular file can be loaded multiple time inside different containers. However this file can be allowed only in some container and not in all of them, but since IMA measure that file only the first time, then we are not aware that this file is in execution on multiple containers. In order to resolve that problem, the IMA module has been changed by adding two kernel boot parameters that enables the deactivation of his internal caching mechanism.

In conclusion, the proposed solution by this thesis enables the Remote Attestation process on Docker v1.10+, and resolves some issues found during the previous development of the project. It also extends the IMA module by enabling to control each cache system independently. The solution is then compared to similar solutions available in the market.