Permanent verification of integrity and authenticity of files in an Infrastructure as Code

Bachelor of Applied Sciences Thesis
Sergio Guarino

Abstract

The emergence of virtualization technologies and cloud computing made it possible to create IT infrastructures made up of hundreds or thousands of servers. Their deployment, management and configuration are often done through network automation tools (aka Infrastructure as Code, or IaC), that allow for quick and easy operations compared to manual ones.
Automation makes the servers also subjects to constant changes and ensuring the security of the applications hosted on them is of fundamental importance, even more so when the services are made publicly available. The purpose of this work is to conceive and implement a security solution compatible with the existing network automation tools and capable of providing constant verification of the integrity and authenticity of files located on an infrastructure made up of servers running Debian OS.

The first step to approach the problem was to make a comprehensive study of the state of the art of automation tools and of methods for providing integrity and authenticity of files. After the initial analysis, two tools were chosen: Ansible as automation engine and FS-Verity as verification mechanism. FS-Verity is a Linux kernel functionality that verifies the integrity of the targeted files each time they are opened, thus allowing a constant integrity check. In addition, it provides an authenticity mechanism based on digital signature, ensuring that only trusted parties can manipulate files.

The in-depth study of these two tools brought to the design of a security architecture in which Ansible employs FS-Verity functionalities to protect a potentially unlimited number of files, while FS-Verity ensures the integrity and authenticity check in background. A monitoring service was also included in the design, to add log entries in case of a security breach and to allow the interaction with infrastructure monitoring tools.

This architecture resulted in the development of a custom Ansible Module that would effectuate operations in an idempotent way, meaning that it would not try to activate the protection on already protected files. This is particularly important in infrastructures with several thousands of files to protect.
Other than the Ansible Module and the monitoring service, another component developed was a tool that allows the deactivation of FS-Verity on selected files, since by design, once activated, FS-Verity can't be disabled anymore.

The projec was finally tested in a small scale infrastracture (as a big one was not available) and it successfully operated in the way it was designed and developed.