Access Control Management and Orchestration in NFV Environment

Tran Quang Thanh [1, 2], Stefan Covaci [2], Marius Corici [1] and Thomas Magedanz [1]
[1] Fraunhofer Institute FOKUS, [2] Technical University Berlin Germany

***Abstract*—Network Functions Virtualization (NFV) represents a very large paradigm shift in the development and the deployment of network services. In this paper, a security architecture for NFV environment is presented concentrating on the automation of access control management. As an illustration, a testbed and a demo application have been implemented and deployed based on open source solutions.**

***Keywords*—Access Control, MANO, NFV, Security**

I. Motivation

When a new network component is added to the system, in the current situation, many security solutions (e.g. firewall, API gateway) need administrators to adapt the access policies accordingly. This is not feasible in an NFV environment where the network functions are changing continuously. An automation approach is required to consider such changes without the need of intervention from the administrators. In the first two releases, ETSI NFV MANO specifications [1] cover only the performance management and fault management for network services. In the latest version, security management integration is discussed and being improved [2]. Some important ideas have been introduced including the security management framework (Figure 1) and the concept of security management lifecycle with three main stages (e.g. Security Planning, Enforcement and Monitoring).

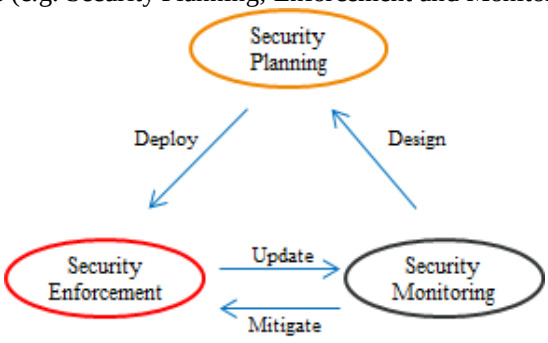

Figure 1. ETSI NFV Security Lifecycle Management

II. Framework Implementation

Our NFV security architecture (Figure 2) takes into account the current ETSI MANO security management high-level framework [2] to provide a flexible and fine-grained access control protection for NFV-based (network) services.

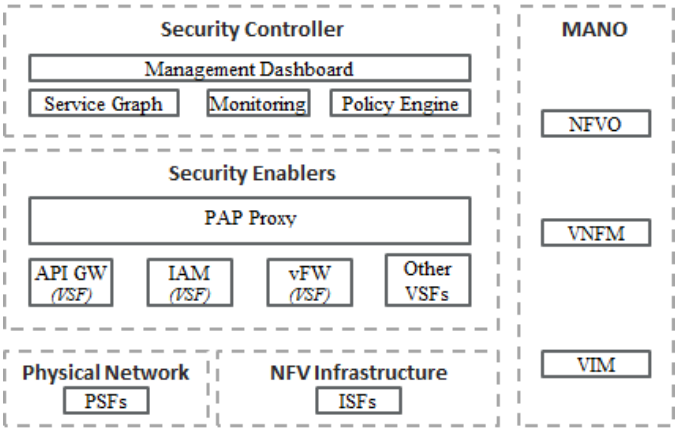

Figure 2. High-level NFV Access Control Management Architecture

In order to support the need of automated security management, the Security Controller (SC) is proposed with four main functional components: Management Dashboard, Service Graph, Monitoring and Policy Engine. Different "Security Enabler" software components (e.g. Virtual Firewall, API Gateway, Identity and Access Management) are included to support different types of service access control requirements. The management and monitoring of such security functions are provided by associated "PAP Proxy" Enablers. Such enablers can not only support configuration of all security required including Virtual/Infrastructure/Physical Security Functions (VSF/ISF/PSF) but also help to collect specific monitoring information from associated security function enablers.

The Security Controller manages two types of policies: Access Control and Decision-Making. The former are enforced by the corresponding "Security Function" Enablers. The latter, Event-Condition-Action (ECA) policies, are required to ensure the consistent of the former across application life cycle.

Current Testbed and Demo Service:

A NFV testbed has been set up with MANO functionality provided by our open source OpenBaton toolkit [3]. OpenBaton implements the current ETSI NFV MANO standard and is designed flexible enough (e.g. modular architecture, interoperable with multiple clouds, different VNFM vendors) to support different research requirements. OpenBaton uses OpenStack [4] to provide virtualized

ISBN 978-3-901882-94-4

infrastructure, and Zabbix [5] for VNF monitoring. A prototyping of Security Controller and a demo application have been developed. Spring Boot framework, a production-grade framework to develop Java applications and services, was selected to implement most of demo components. Other open source solutions are utilized such as FIWARE Orion Context Broker [6] or Netflix Spring Cloud Zuul engine [7]. The demo application is a simple client-server application where access from the client to the server is controlled by a virtual firewall (Figure 3).

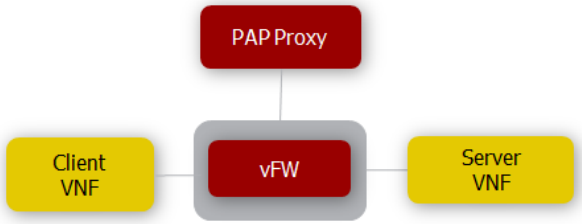

Figure 3. Demo Service Graph

In the demo application, one access control policy was defined that allows any Client VNF to access Server VNF. Several supported ECA policies were also specified including:

IF ["Deployment Finished"].Status [eq] ["SUCCESS"] THEN [POLICY_UPDATE to "PAP Proxy"]
IF ["VNF Scale Out"].Type [eq] ["Client"] THEN [POLICY_UPDATE to "PAP Proxy"]
IF ["VNF Termination"].Type [eq] ["Client"] THEN [POLICY_UPDATE to "PAP Proxy"]

The first ECA policy allows configuring prepared access policies to the associated security function enablers (the vFW in the demo) right after the successful deployment of the demo application. The second and third ones are defined to ensure the consistent of access control policies during application operation to automatically adapt to the changing scenarios (VNF scaling, VNF termination or found unavailable). The demo flow is described further in Figure 4.

Our NFV testbed and the demo application help users to have a practical look and feel about how to achieve automated security management in NFV environment. Specifically, three security requirements have been addressed and illustrated:

Embed Security Functions (Security protection could be automatically and transparently embedded in a virtual network infrastructure as virtual or NFVI-based security functions).

Security Management Lifecycle Support (Security policy enforcement adapts across application lifecycle including instantiation, scaling and termination).

Dynamic Security Incident Response (Security policy enforcement adapts in case of unforeseen events such as DoS attack, fault).

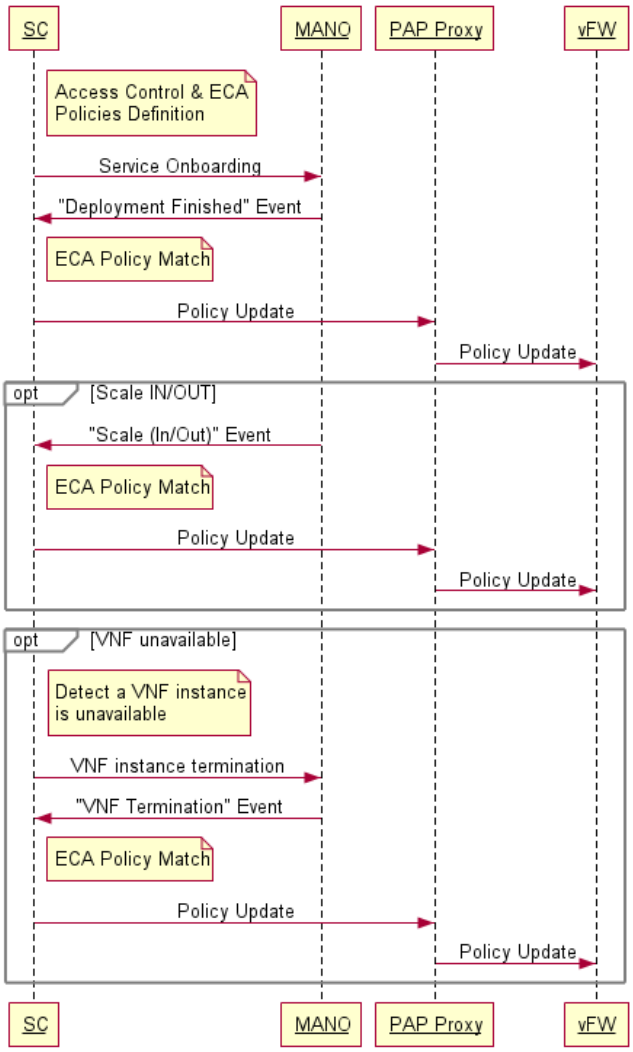

Figure 4. Demo Sequence

III. Conclusion and Future Work

In this paper, security architecture and demo applications for NFV environment are described concentrating on the automation of access control management. Other security supported technologies such as SDN-based Service Function Chaining are under investigation to integrate into our work.

References

[1] ETSI NFV MANO specification, http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf
[2] ETSI NFV Security Management and Monitoring, http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/013/03.01.01_60/gs_NFV-SEC013v030101p.pdf
[3] OpenBaton, http://openbaton.github.io
[4] Open source software for creating private and public cloud, https://www.openstack.org
[5] Zabbix: The Enterprise-class Monitoring Solution for Everyone, http://zabbix.com
[6] FIWARE Orion Context Broker, https://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker
[7] Spring Cloud Netflix, http://cloud.spring.io/spring-cloud-netflix/spring-cloud-netflix.html